POPCOM BOOKS

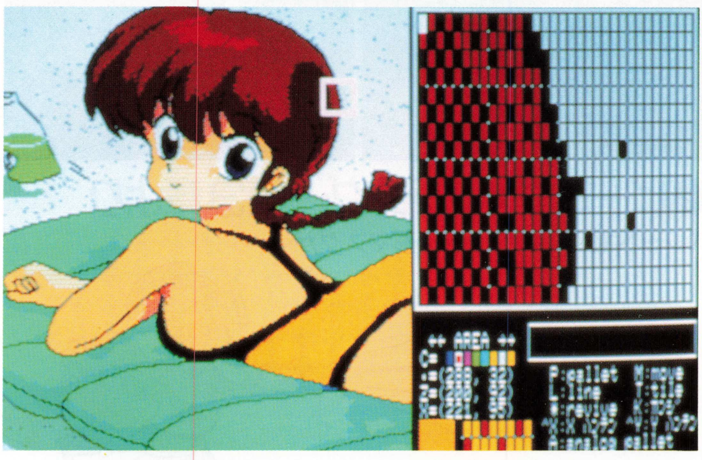
PC-8801シリーズ

マシン語ゲームグラフィックス

日高 徹・著

小学館

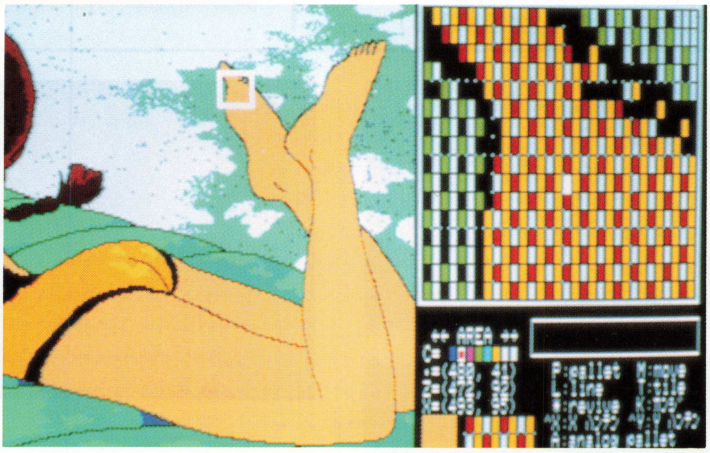
ワ！ 大型グラフィックスができる！



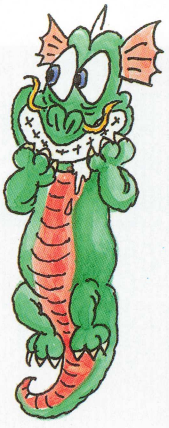
●グラフィック・ツール『PMAN88』により、フル画面のエディットが自由自在！
●原画の入力は白黒／カラースキャナ対応の『SCAN88』で簡単！
●エディットした絵は圧縮をかけてセーブできるのでゲームへの応用もラ〜クラク！
●（本文第5章『魅惑の大型グラフィックス』より）



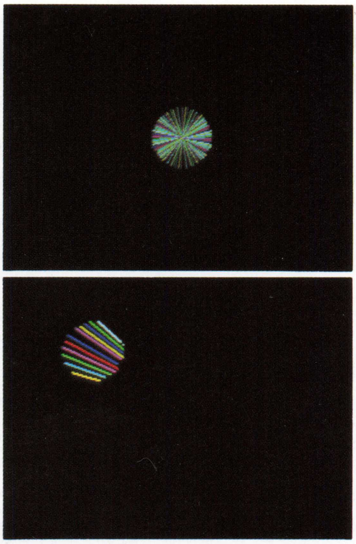
●アララ、画面がスクロールした……なんて驚かないで！　エディット中に←→キーを押すと、画面がスルスルと……。



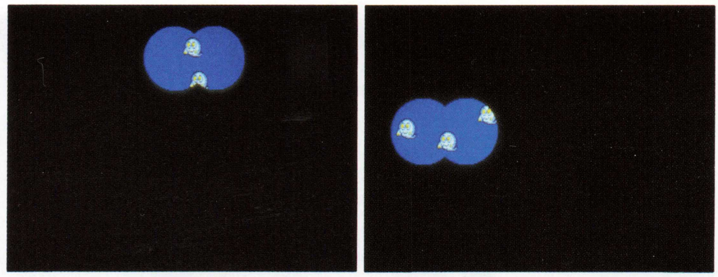




●第3章より



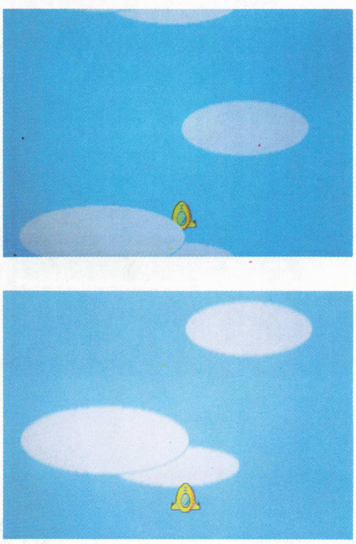
テキスト・マスクでキラ〜クにスポット処理(第3章より)



●ピーヨウォッチング…第3章より

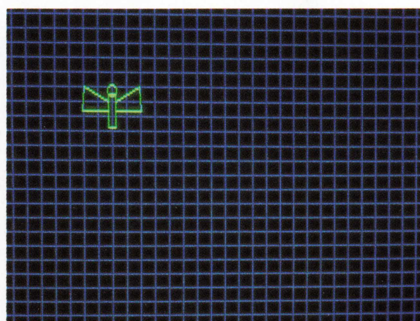
パレット操作により、不定形マスクが……

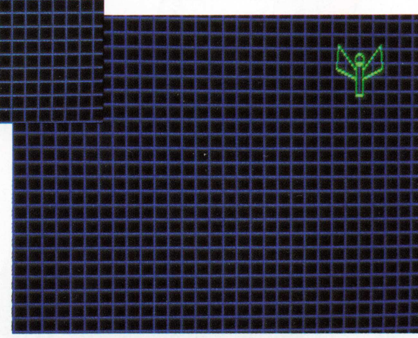


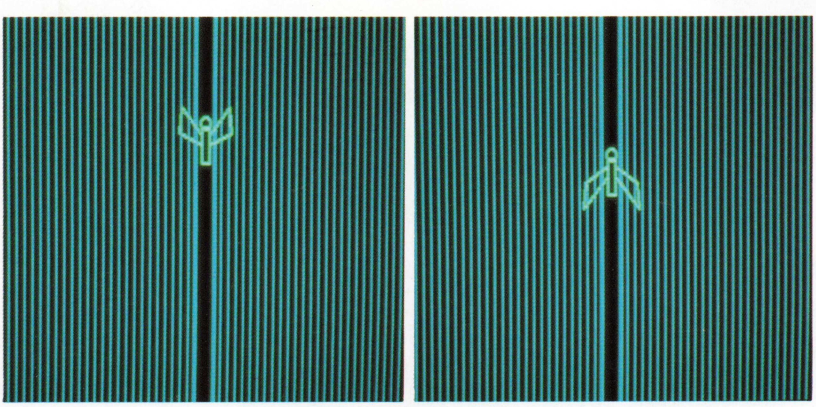


●第3章より

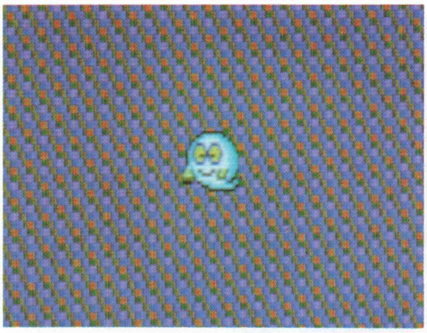
パレット操作によるプレーン分割のいろいろ







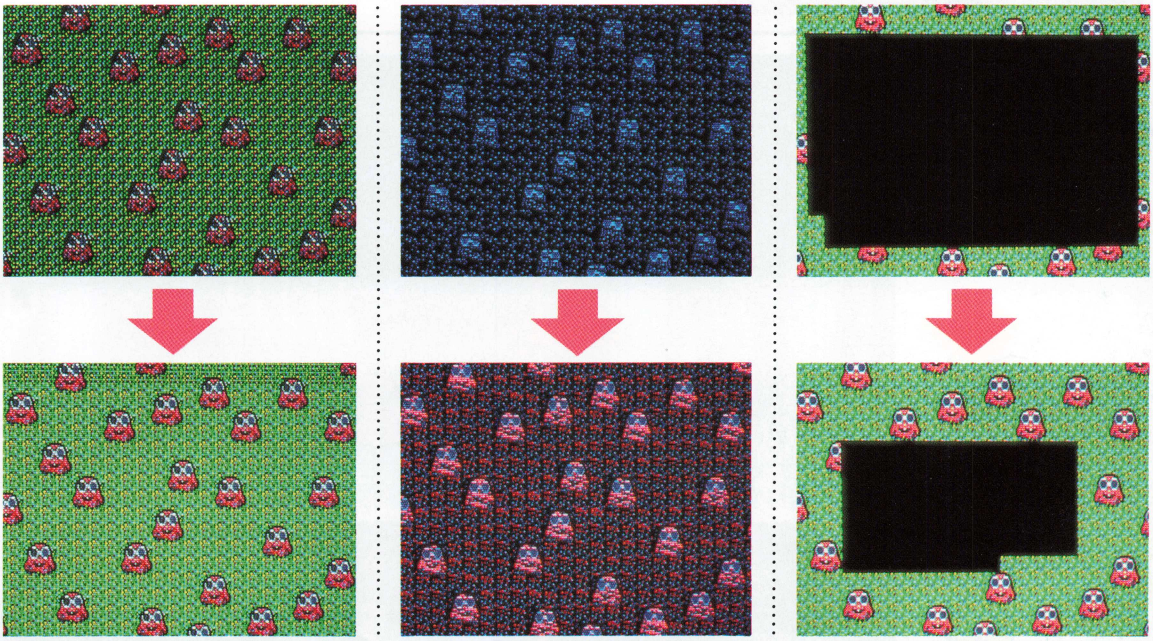


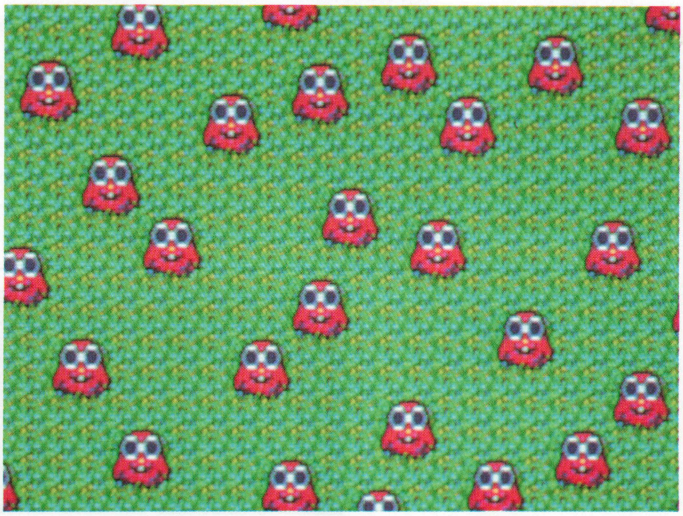




●第2章より

おばけのピーヨがチカチカと反転





●第3章より

おしゃれな画面表示／消去も、工夫しだいでいろいろできる！





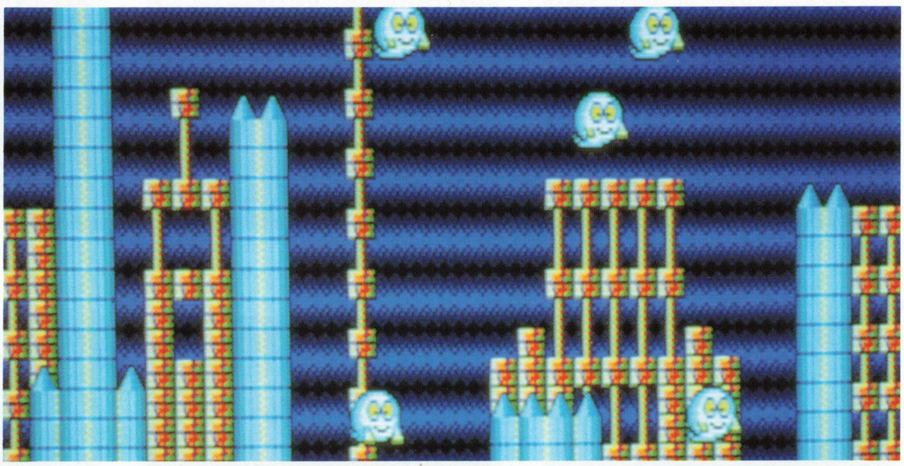
●第3章より

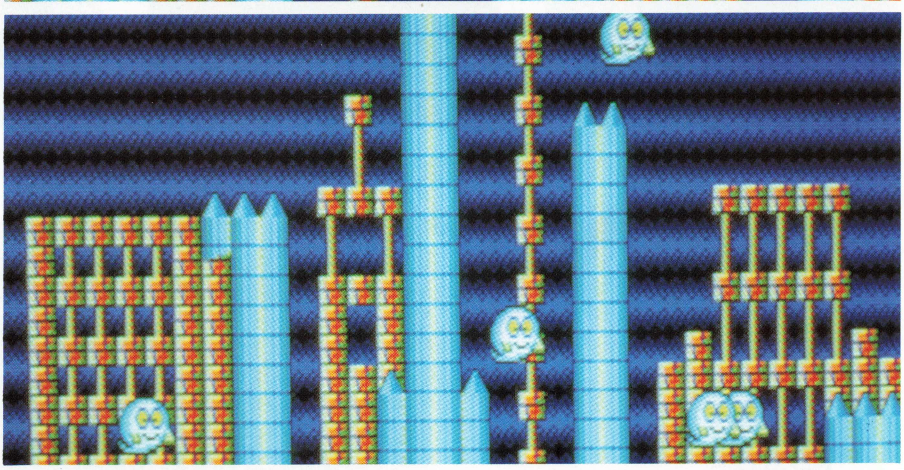
カッコイイ漢字縮小表示のすべて……



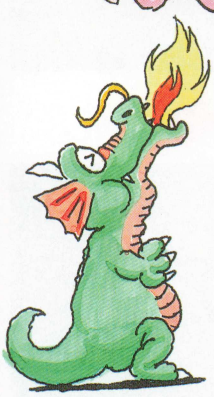
●第4章より

最新の多重スクロールテクニックを大公開しちゃいます！



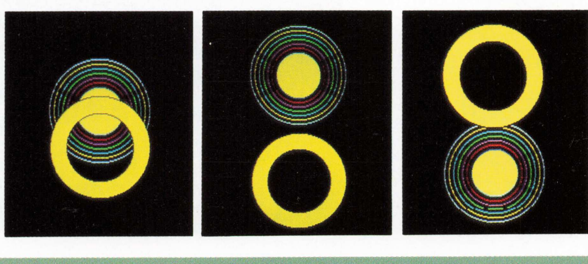






●第3章より

重ね合わせアニメーション処理とは？



PC-8801シリーズ

マシン語ゲームグラフィックス

小　学　館

はじめに

コンピュータの世界では、マシン語以外の言語（BASICやFORTRANなど）を「マシン語に比べれば人間の言葉に近い」ということで高級言語と称していますが、実際の人間の言葉というものは常に変化し進化するものです。例えば流行語などはその典型で、次から次へと生まれてははかなく消えていきます。

このように流動的な人間の言語に比べ、コンピュータにおける高級言語は変化や進化とはほとんど無縁のものです。もちろん、機種による違いや部分的に拡張されることはありますが、ユーザーが独自に言語を発展させることは不可能な状況です。つまり、BASICで書かれたプログラムは5年前のものでも古さを感じることはなく、いわゆる文語体／口語体のような区別をつけるのは困難です。

一方、マシン語というのは単体ではアルファベットと同じですから、組み合わせて初めて言語としての役割を果たすものです。一見すると原始的に思えますが、人間の言葉もルーツをたどればマシン語以上に原始的です。だからこそ、変化し発展することができるわけです。マシン語が言葉になるとき、人はそれをテクニックと言います。自分自身の技術的成長もありますが、こういったテクニックというものは時代とともに変化し進歩していくものです。そこには明らかに流行もあるし、過去形となった旧式のテクニックもあります。こう考えると、まさにマシン語こそが真の高級言語なのかもしれません。

そして、こういった流行に最も敏感に反応するのがゲームにおけるテクニックです。中でも、グラフィックスはゲームの顔であり象徴ですから、常に最新の技術が求められており、またプログラマーの腕の見せどころでもあります。数年前に発刊した『マシン語ゲームプログラミング』（アスキー）のテクニックが、ことグラフィック操作に関する限り色あせてくるのもやむを得ないことでしょう。

これまでに多くの読者の方々から、様々な要望や激励のお便りを戴いていますが、今回は特に希望の多い最新のグラフィックス・テクニックを集め、できるだけわかりやすく解説したつもりです。ゲームそのものの謎解きも楽しいでしょうが、ゲームに隠されたマシン語の謎について考えるのは、一段上の知的ゲームです。本書が、そのマシン語の謎を解決する一助になれば筆者としてうれしい限りです。

また、グラフィックスを語る上において、グラフィック・ツールの存在を無視することはできません。本書では、グラフィックス開発用支援ツールとして、大型グラフィックス用に『PMAN88』、キャラクタ・パターン用に『PTER88』、そしてスキャナ

コントロール用に、『SCAN88』という3つのツールを用意しました。これらは私がプロ用に開発したバージョンを一般用に使いやすくリメイクしたものです。市販商品のように実用から離れたハデな部分はカットしましたが、ゲームを作る上においてはいずれも十分な実力を備えているものばかりです。きっと、あなたのPC-8801にBASICでは味わえなかった楽しい世界が広がることでしょう。

なお、本書執筆にあたり日本電気株式会社、オムロン株式会社よりデモ機材等の貸し出しを受けましたことを、この場より改めて御礼申し上げる次第です。

1990年3月　　日 高　徹

目　次

第1章　グラフィックスの基礎

コンピュータと人間を結ぶ最大のインターフェース、それはディスプレイ装置（標準的にはブラウン管）による画面表示です。もしも、コンピュータが画面に何も出力してくれなかったなら、われわれはコンピュータをどのように利用していいか……。プログラムを組むことはおろか、ゲームで遊ぶこともできずに右往左往するばかりでしょう。

普段、無意識のうちに操作していることすべてが、実はディスプレイに表示するというプログラムのお陰で成立していたのです。つまり、画面表示はあらゆるプログラムの窓であり、けっして避けては通れない基礎テクニックなのです。本当の主役は画面に現れないプログラムでも、第三者にとっての主役は常に画面です。まさに「画面を制する者がすべてを制す」と言っても過言ではないでしょう。

そのための第一歩、それはPC-8801シリーズにおける画面表示の能力を正確に把握することです。市販ゲームの高度な画面処理も、すべてはその範囲の中でテクニックを駆使した結果なのですから……。



1-1 表示画面の種類

初代のPC-8801が「究極の8ビットコンピュータ」として発売されたのは1981年のことでした。その後、幾度もモデルチェンジが重ねられましたが、画面表示の基本部分に関してはまったく変わることなく進化しています。もちろん、表示処理速度のアップやカラーパレットのアナログ化（512色中から8色選択可能）など、細かな改良点はアチコチにあります。しかし、画面を構成しているVRAM（ビデオラム）*の構造は、昔も今も同じように設計されているのです。

すでにご存じの方も多いと思いますが、とりあえずPC-8801シリーズにおける画面構成を確認することにしましょう。図1-1-1を見てください。

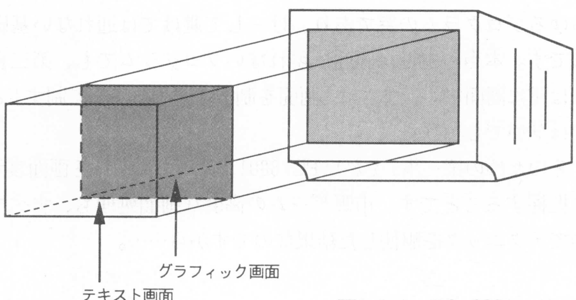


図1-1-1　PC-8801シリーズの画面構成

この図から、PC-8801にはテキスト画面とグラフィック画面とがあり、テキスト画面のほうが表示優先順位が高いことがわかります。つまり、グラフィック画面に絵が表示されていても、テキスト画面に文字が表示されれば、文字の部分だけ絵が隠れてしまうということです。また、グラフィック画面にはカラーモードと白黒モードがあり、BASICでは次のように使い分けることができます。

●カラーモード：

640×200ドットで1画面（SCREEN 0）

●白黒モード：

640×200ドットで3画面（SCREEN 1）

640×400ドットで1画面（SCREEN 2）

PC-8801にはグラフィック画面として3枚のプレーン*が用意されており、実際

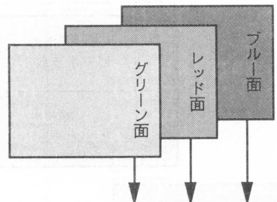
VRAM（ビデオラム）：

その中身が直接画面表示に関係する特殊なメモリ。画面に文字を出したり、絵を表示したり……と、画面はこのビデオラムに入れられたデータによって創られている。

3枚のプレーン：

正しくはグラフィック・ビデオラム（G.VRAM）0～2と呼ばれるが、本書ではカラーモード／白黒モードを問わずB面／R面／G面という表記で区別をしている。これは、カラーモードではこのほうがわかりやすいので、白黒モードでも統一したほうがプレーンを一致させやすいからである。

にはこの３枚のプレーンをモードによって使い分けています。標準的なカラーモードでは、３枚のプレーンをＢ（ブルー）面、Ｒ（レッド）面、Ｇ（グリーン）面とに分け、それを組み合わせることでカラー８色表示を実現しています（図1-1-2)。



G	R	B	カラー番号
0	0	0	0
0	0	1	1
0	1	0	2
0	1	1	3
1	0	0	4
1	0	1	5
1	1	0	6
1	1	1	7

0：そのプレーンが発色しない
1：そのプレーンが発色する

図1-1-2　グラフィック画面のカラー発色の仕組み

これに対し、白黒３面モードというのはこの３枚のプレーンをそれぞれ独立したプレーンとして使うモードであり、各プレーンを個別に直接アクセスできるマシン語においては特に必要がありません。というのは、表示プレーンの選択はパレットによっても可能だからです。例えば、Ｂ面の内容を見たければ、Ｂ面の関わっているカラー＝１（他のカラーでもよい）とし、Ｂ面の関わっていないカラー＝０とすればいいのです。

パレット指定：
0→0，1→1，2 →0，3→1，4→0，5→1，6→0，7→1

この方法であれば、特定のプレーンを好みのカラーで見られるばかりでなく、面に優先順位をつけて合成するということも可能です。あえて白黒３面モードにしないほうが、用途的にも広いといえるでしょう。

（例）Ｂ面（カラー７で表示）とＲ面（カラー１で表示）を合成する。表示はＢ面を優先する（合成してダブる部分はカラー７となる)。
パレット指定：
0→0，1→7，2→1，3→7，4→0，5→7，6→1，7→7

なお、白黒１面（640×400ドット）モードは、ハードウェア的に特殊な構造に切り換えられてしまうので、縦200ラインモードとの互換性はありません。このモー

ドではB面として使用されているビデオラムで画面の上半分を表示し、R面として使用されているビデオラムで画面の下半分を表示します。G面は使用されません。

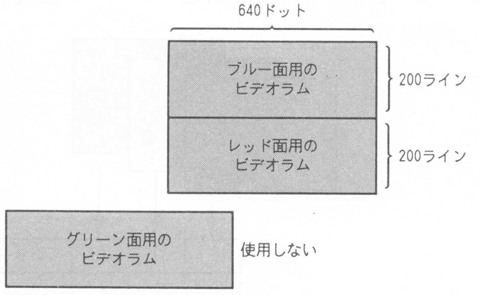


図1-1-3　縦400ラインモードにおけるプレーンの状態

これらの白黒画面（3面と1面）モードにおいては、テキスト画面との合成がカラーモードとは若干違っています。テキスト画面の文字が正しく表示されるのはグラフィック画面の黒い部分（描かれていない部分）だけで、白い部分（描かれている部分）はテキスト文字のカラーで着色されてしまうのです。したがって、縦400ラインモードではグラフィック表示部（白い部分）のテキスト文字は見えなくなります。文章ではわかりにくいかもしれませんが、この事実は BASICでも簡単に確認できますから、白黒モード（SCREEN 1またはSCREEN 2）で実験してみてください。

では、次にテキスト画面とグラフィック画面のVRAM（ビデオラム）がメモリのどこに存在しているのか、メモリ全体のマップを見ながら確認することにしましょう。

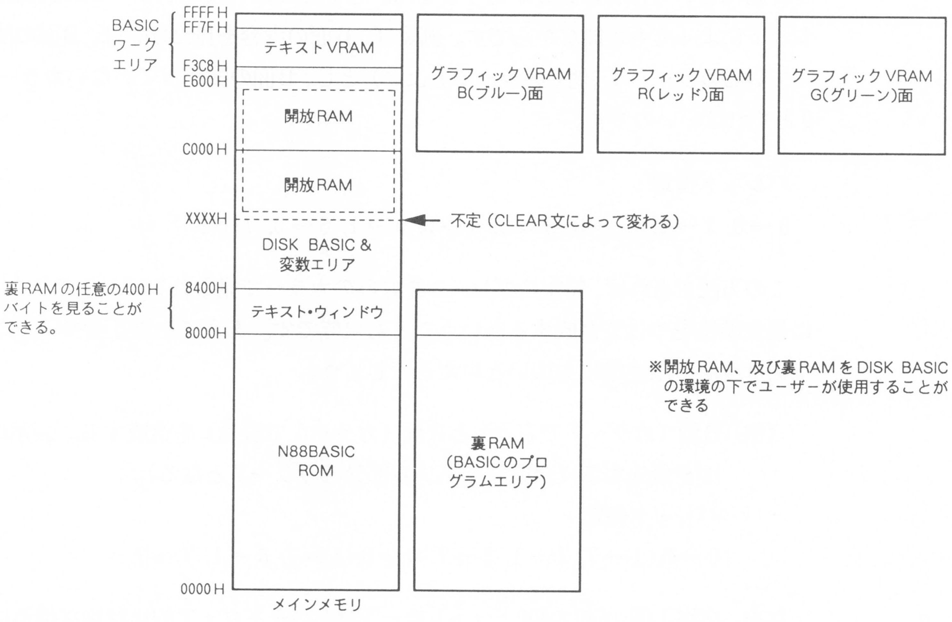


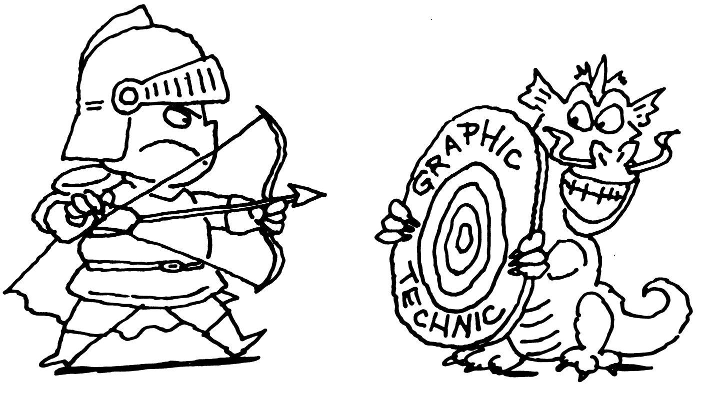
図1-1-4　PC-8801の基本的なメモリマップ

図1-1-4から、テキスト画面はBASICワークエリアの一部としてメインメモリ上にあることがわかります。また、グラフィック画面は3プレーンがそれぞれメインメモリから独立した状態で存在しており、バンク切り換え*という手法によって使用可能となります。ここで注意しなければならないのは、バンク切り換えによってグラフィック画面を選択した場合、その時点で同じアドレス（C000H～FFFFH）にあるメインメモリは切り離されているということです。つまり、C000H以降にあるプログラムでバンク切り換えができないのは当然ですが、バンク切り換え中に使用するワークエリアやスタックエリアもC000H以前に確保しておかなければなりません。

なお、ここで示されているメモリマップは、ユーザーがマシン語プログラムを組むことを前提にしており、すでにCLEAR文によって開放されたRAMエリアが確定しているものとします。CLEAR文を実行していない場合、E500H番地台の後半にスタックエリアが存在している上、図1-1-4でXXXXHとなっているアドレスが確定しないので、安心してマシン語プログラムを組むことができません。

また、PC-8801mkII SR（以下PC-8801SRと略す）以降の機種やサウンドボードIIを取り付けた機種においては「NEW CMD」あるいは「BLOAD"fm.ipl", R」を実行することにより、ワークエリアの内容が変わります。このことは開放されるRAMの状態が変化することを意味しますから、注意してください。

それから、本書のプログラムは一部を除いて割込み禁止の状態で実行します。ユーザー割込みを使ってサウンドと組み合わせたりすることも可能ですが、それについて興味のある方は拙著『マシン語サウンド・プログラミング』（アスキー発行）のほうをご覧ください。本書はあくまでもグラフィック・テクニックがターゲットです。



バンク切り換え：

列車の線路を切り換えるように、使用するメモリバンク（メモリ層）を必要に応じ切り換えること。8ビットのCPUでは一度に64Kバイトしかアクセスできないため、このような手法でそれ以上のメモリを扱えるようにしている。

1-2 画面操作のコモンセンス

すでにマシン語でプログラムを組まれている方にとって、この程度の内容は知っていることばかりと感じるかもしれません。しかし、市販されているゲームの高度なテクニックも、その知っていることを応用しているに過ぎないのです。基礎を振り返ることで、新しい何かを発見することが往々にしてあります。知識をリフレッシュするつもりで、テキストVRAMとグラフィックVRAMの初歩的な使い方を確認しておくことにしましょう。

テキストVRAMは図1-1-4で示されていたように、メインメモリのF3C8H〜FF7FHを占めています。これをもう少し詳しくしたのが図1-2-1です。

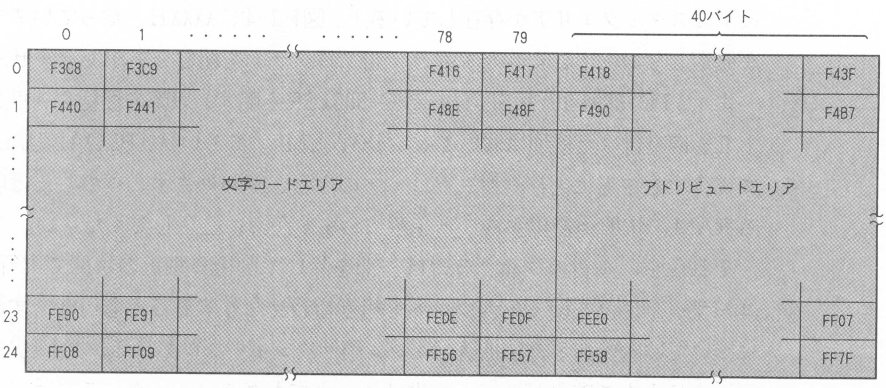


図1-2-1　テキストVRAMのアドレスマップ

テキスト画面の桁数（横の文字数）が80（WIDTH 80）に設定されている場合、文字コードエリアの1バイトは画面のアスキーコード1文字に対応します。桁数が40の場合は、偶数アドレスのみが画面に対応し、奇数アドレスは意味を持ちません。同様に、行数（縦のライン数）が20行のときは、21行目以降のアドレスは使われません。

次に、アトリビュート（属性）エリアについてですが、これはテキスト画面の文字にカラーやブリンクといった属性を付けるためのもので、2バイト1組（1行につき最大20種の変化が可能）で表すようになっています。

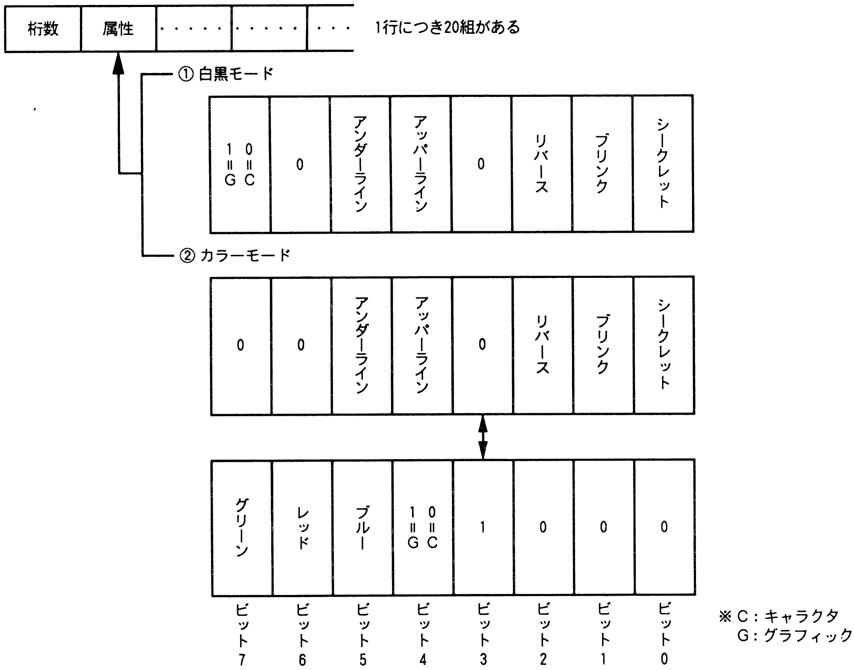


図1-2-2　アトリビュートの内容（2バイト1組）

属性の内容は、白黒モードとカラーモードでは図1-2-2に示されているように同じではありません。また、属性のクリアは、白黒モードでは00H、カラーモードでは00HあるいはE8Hで行います。一方、桁数については50Hでクリアするか80Hでクリアするかによって属性への意味が違うなど、いくつかの注意点があります。

①桁数＝50Hでクリアする場合
　指定された桁数＋1から指定された属性を有効とする。
②桁数＝80Hでクリアする場合
　桁数＝0の場合は①と同様となる。それ以外の場合は、指定された桁数まで指定された属性が有効となる。ただし、最後に指定された桁数＋1からは、属性をクリアするのに使用した属性が有効となる。また、各行の有効となる属性以降はクリアされたままとなっていなければならない。
③桁数の指定は必ず小さいほう（画面左側）から順に行う。
④同一アドレスに対して複数の属性を指定することはできない。

なお、40桁モード（WIDTH40）のときには桁数を倍にして数えます。桁数や属性については、意外と予想と違った結果になるときがあるので、プログラムで直接操作する場合はモニタのEコマンドで確認しておくと安全です。

ここで属性のところにあるグラフィックについて少々説明を加えておきます。これは正確にはテキスト画面によるロー・レゾリューション・グラフィック（160×100ドット）のことで、その昔PC-8801の前身であるPC-8001（併売されていた

時期もある）において使用されていたN-BASICモードのグラフィックと同じものです。640×200ドットのフルカラー・グラフィックに比べるとオモチャのような存在ですが、グラフィック画面と簡単に重ね合わせることができるので、いろいろな用途があります。具体的な活用例はこの先折りに触れ紹介しますので、ここではテキスト・グラフィックの存在とビット別の内容を覚えておいてください。

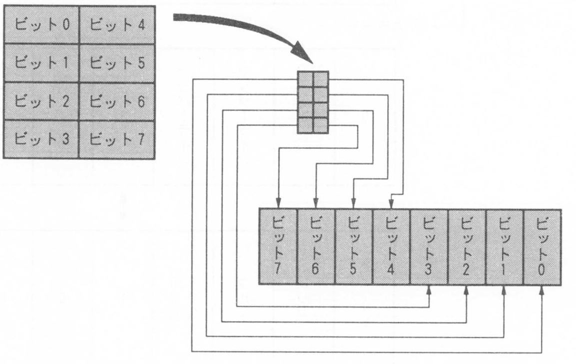


図1-2-3　テキスト・グラフィックのビット別内容

では、次にグラフィック画面ついて、ビデオラムのアドレスマップとドットとの関係を見てみましょう。

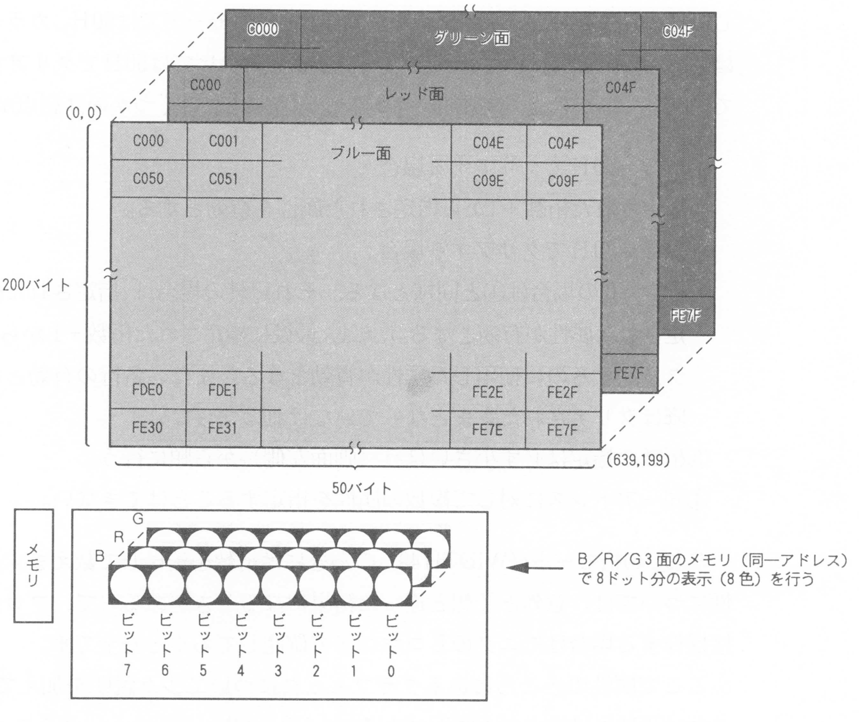


図1-2-4　グラフィックVRAMのアドレスマップとドットとの関係

図1-2-4に示されているように、グラフィック画面はBRG3プレーンの同一アドレスにおける1バイトで8ドットを表します。したがって、8ドット単位でデータをセットする場合は単純ですが、ドット単位で操作する場合には他のドットに影響を与えないようプログラム側で考慮しなければなりません。例えば（0,0）の位置にカラー5で点を打つ場合、画面に何も表示されていなければB／G面のC000Hに80H（10000000B）を入れればいいのですが、何か絵が表示されているときはB／G面（C000H）のビット7だけをセットし、R面のビット7をリセットするようにしなければならないわけです。

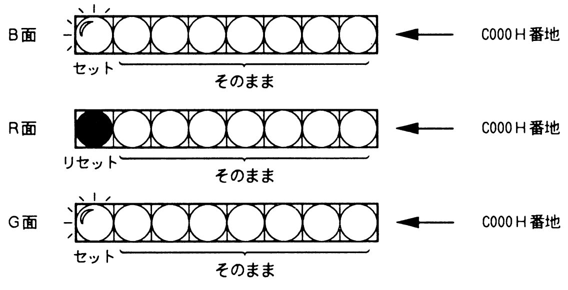


図1-2-5 （0,0）にカラー5の点を打つ

これを実際にプログラム化したのがLIST 1.1です。バンク切り換えの方法は、マシン語をかじった経験があれば誰でも知っていることと思います。また、このプログラムからでも簡単にわかりますのであえて解説はしません。ここでは、各プレーンともC000H番地のビット7をAND命令でリセットし、その後でカラーに合わせてOR命令を実行しているということに注目してほしいのです。

あるいは、こんな面倒なことをしないで、SET／RES命令で直接メモリを操作すればいいと思われる方もいるでしょう。しかし、カラーや座標（1バイトの中での）の変化に簡単に対応するためには、レジスタの値を引数として利用できるようにしておくほうがプログラム効率の面で優れているのです。



LIST 1.1

```
                         ORG     0B700H

B700 F3                  DI                    ・割込み禁止
B701 0E05                LD      C,5           ・C＝カラー番号
B703 167F                LD      D,01111111B   ・D＝ドットリセット用データ
B705 1E80                LD      E,10000000B   ・E＝ドットセット用データ
B707 2100C0              LD      HL,0C000H     ・HL＝G.VRAMアドレス
B70A D35C                OUT     (5CH),A       ┐
B70C CD1DB7              CALL    DOTON         │
B70F D35D                OUT     (5DH),A       │
B711 CD1DB7              CALL    DOTON         ├・指定カラーで点を打つ
B714 D35E                OUT     (5EH),A       │
B716 CD1DB7              CALL    DOTON         │
B719 D35F                OUT     (5FH),A       ┘
B71B FB                  EI                    ・割込み許可
B71C C9                  RET                   ・BASICへ戻る

B71D 7E         DOTON:   LD      A,(HL)        }・G.VRAMの値を取り出しビットリセット
B71E A2                  AND     D
B71F CB09                RRC     C             ┐
B721 3001                JR      NC,DTSET      ├・カラーに合わせビットセットをする
B723 B3                  OR      E             ┘
B724 77         DTSET:   LD      (HL),A        }・G.VRAMにデータを入れリターン
B725 C9                  RET
```

ここでは単に点１つを打ったに過ぎませんが、実はこれは規模は小さくても「重ね合わせ」そのものです。最近のゲームでは重ね合わせ処理など当然のテクニックですが、その基本形がこの『ANDを取ってORを取る』ということなのです。つまり、ANDで存在しているものをいったん消去し、ORでそこに描き込むというわけです。

たかだかドット１つのプログラムですが、その延長線上にはあらゆる高度なテクニックが待ちかまえているのです。ちょうど、すべてのグラフィックスがドットの集まりであるかのように…。

なお、本書のプログラムは筆者が開発したエディタ／アセンブラ『HIT-88』の文法に基づいて記述されています。『HIT-88』の入手方法は巻末に掲載してあります。その他のアセンブラを使用される方は、特殊命令などの文法に違いがないかどうか、一通り確認してください。添付のディスクには、『HIT-88』用のソースリストとアセンブル済みのマシン語プログラムがBSAVEされていますので、プログラムはアセンブルすることなく実行／確認することができます。なお、ファイルネームは、LIST1.1の場合“L=1-1”、LIST1.2は“L=1-2”となっています(他も同様です)。プログラムの実行は今後も含めて原則としてすべて次のようになります。『HIT-88』でアセンブル直後に実行する場合はCLEAR命令は不要ですが、ディスクからBLOADする場合はBLOADの前に必ずCLEAR命令を実行してください。

```
CLEAR, &HB5FF
A=&HB700:CALL A
```

今回は、実行後自動的にBASICへもどって来ますが、ループになっている場合は［ESC］キーによってBASICへもどるようになっていますので、併せて覚えておいてください。

1-3 テキスト画面の豆知識

手品とマシン語プログラム。一見すると何の脈絡もなさそうですが、実はちょっとした共通点があります。例えば、ごく普通のトランプを使った簡単な手品でも、タネがわからなければスゴイものに見えることがあります。同様に、マシン語プログラムにも相手の盲点をつくようなトリックがあるのです。もちろん、タネがわかれば「ナァ〜ンダ!!」というようなことであるのは事実です。しかし、これが高度なテクニックと組み合わせられると、ついつい「もしかすると手品ではなく本物の魔法では……!?」などと思ってしまうから不思議なものです。

市販ソフトの中には、まるでPC-8801が特別な機能を隠していたかのごとくに、見事なテクニックを披露してくれるものもあります。そういった見事なものは、たいてい力任せの本格プログラムですが、その裏には常に基本のトリックが潜んでいます。そして、テキスト画面というのはそういったトリックにはうってつけの存在なのです。

テキスト画面とアトリビュートの関係は前節で解説しましたが、テキスト画面全体をコントロールするにはそれだけでは不十分です。といっても、CRTC (CRTコントローラ)*についての詳しい知識が必要というのではなく、テキスト画面のオン／オフの方法を知る程度ですが……。

PC-8801〜PC-8801mk IIでは、よくDMAをストップ（出力ポート51Hに0を出力）して実行スピードのアップを図っていました（詳しい用例は『マシン語ゲームプログラミング』参照)。しかし、PC-8801SR以降の機種からは、テキストDMAによるCPUの一時停止がなくなり、スピードアップ目的でのDMAストップは不要になりました。つまり、テキスト画面がいつでも自由に使用できるということです。

単にテキスト画面のオン／オフであれば、DMAのストップではなく操作の簡単な出力ポート53Hを利用するほうが賢明です。出力ポート53Hの内容は、図1-3-1のようになっています。

CRTC (CRTコントローラ)：
テキスト文字の表示を行うためのコントローラ。テキストVRAMに格納されている表示データをDMA (Direct Memory Access) により直接リードし文字表示を行っている。標準スピードモード（V1S及びPC-8801〜PC-8801mk II）においては、このためにCPUが一時停止し全体のスピードが遅くなる。

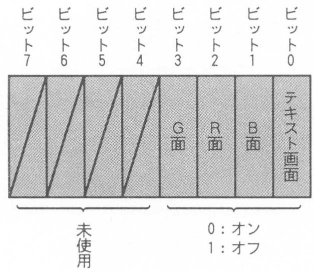


図 1-3-1　出力ポート53Hの内容

このポートはテキスト画面だけでなくグラフィック画面のプレーン別オン／オフができるようになっていますが、これは白黒3面のときにだけ有効でカラー・グラフィックのときには意味を持ちません。したがって、通常はテキスト画面のオン／オフに利用されています。

●テキスト画面のオン

```
XOR  A
OUT  (53H),A
```

●テキスト画面のオフ

```
LD   A,1
OUT  (53H),A
```

ところで、テキスト画面をオフにするだけであればDMAストップも出力ポート53Hを利用するのもプログラム的な差はありませんね。では、なぜ出力ポート53Hがいいのでしょう。その理由は、DMAはオフにするのは簡単でもオンにするのが大変だからです。

では、どの程度大変か……。ちょっとした実験をしながらこの事実を確認してみましょう。とりあえずはLIST 1.2を実行してください。

LIST 1.2

```
                        ORG     0B700H

B700 F3                 DI                        ・割込み禁止
B701 DB09     TLOOP:    IN      A,(9)             ┐
B703 07                 RLCA                      ├・[ESC] が押されていればTOBSCへ
B704 3015               JR      NC,TOBSC          ┘
B706 07                 RLCA                      ┐・[SPACE] が押されていなければTLOOPへ
B707 38F8               JR      C,TLOOP           ┘
B709 3AC0E6             LD      A,(0E6C0H)        ┐
B70C EE01               XOR     00000001B         ├・テキスト画面を80桁／40桁交互に変える
B70E 32C0E6             LD      (0E6C0H),A        │
B711 D330               OUT     (30H),A           ┘
B713 DB09     SPCON:    IN      A,(9)             ┐
B715 07                 RLCA                      │
B716 07                 RLCA                      ├・[SPACE] が押されていなければTLOOPへ
B717 30FA               JR      NC,SPCON          │
B719 18E6               JR      TLOOP             ┘
B71B FB       TOBSC:    EI                        ・割込み許可
B71C C9                 RET                       ・BASICへ戻る
```

このプログラムは、テキスト画面の桁数（横の文字数）を80桁／40桁と、スペースキーを押すたびに交互に変えるものです。出力ポート30HのデータはE6C0H番地に保存されており、各ビットの明細は次のようになっています。

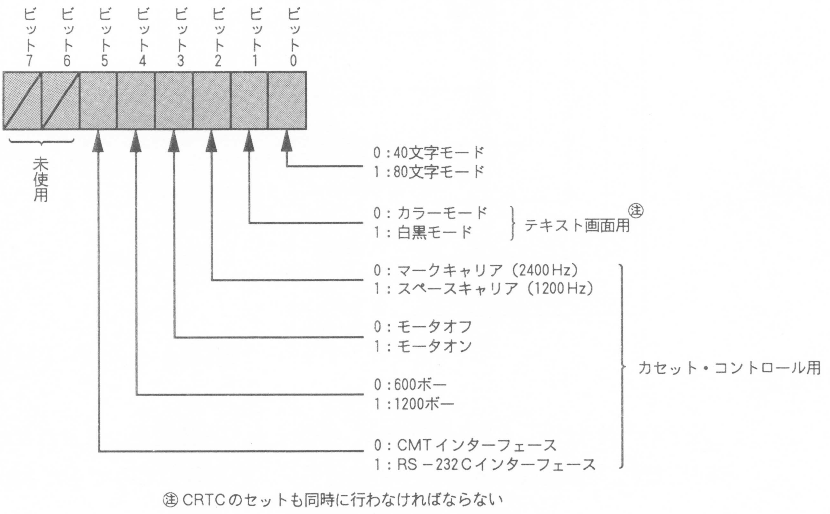


図1-3-2　出力ポート30Hの内容

おそらく、このポートを直接操作する機会はほとんどないと思います。しかし、テキスト画面の桁数を直接変えることにより、前節で解説した「桁数＝40のときには奇数アドレスが無視される」という事実を改めて確認できたはずです。……が、ここで変更になったのは画面上の桁数（文字サイズ）だけであり、モード（WIDTH）そのものが変更になったわけではありません。したがって、この状態では入力した文字が1文字おきに表示されるなど、新種のモードともいえる珍現象が起こります。モードを変えたままBASICにもどって確認してください。80文字モードからだけでなく、40文字モードからも実験してみると逆の珍現象に出会えるでしょう。

では、次にテキスト画面の行数（縦のライン数）を操作してみましょう。行数に関しては、出力ポート31H（データはE6C2H番地に保存されている）のビット5が使えそうな雰囲気なのですが……。

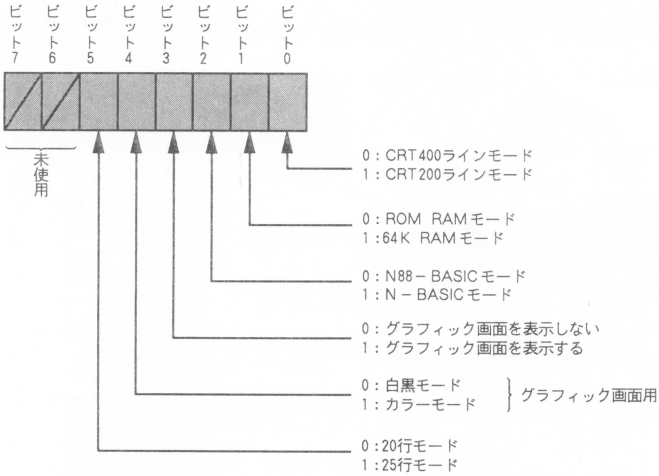


図1-3-3　出力ポート31Hの内容

プログラム（LIST 1.3）の内容は、桁数変更が行数変更になっただけでほとんど同じです。今度は、25行モードと20行モードで実験してみてください。

LIST 1.3

```
                        ORG     0B700H

B700 F3                 DI                          ・割込み禁止
B701 DB09      TLOOP:   IN      A,(9)               ┐
B703 07                 RLCA                        │・[ESC] が押されていればTOBSCへ
B704 3015               JR      NC,TOBSC            ┘
B706 07                 RLCA                        ┐・[SPACE] が押されていなければTLOOPへ
B707 38F8               JR      C,TLOOP             ┘
B709 3AC2E6             LD      A,(0E6C2H)          ┐
B70C EE20               XOR     00100000B           │・テキスト画面の行数指定を変える
B70E 32C2E6             LD      (0E6C2H),A          │
B711 D331               OUT     (31H),A             ┘
B713 DB09      SPCON:   IN      A,(9)               ┐
B715 07                 RLCA                        │
B716 07                 RLCA                        │・[SPACE] が押されていなければTLOOPへ
B717 30FA               JR      NC,SPCON            │
B719 18E6               JR      TLOOP               ┘
B71B FB        TOBSC:   EI                          ・割込み許可
B71C C9                 RET                         ・BASICへ戻る
```

結果はどうだったでしょうか。25行モードから20行モードへ変更すると、行によっては文字の一部が黒いラインで消されてしまいます。その逆に20行モードから25行モードへ変更すると、文字の下にゴーストのようなものが現れます（専用高解像度ディスプレイ使用時）。つまり、行間スペースの表示だけが変更になり、実際の行数は変更されていないわけです。これは、いったいどうしたことなのでしょう。

実は、テキスト画面のモード設定にはCRTC（CRTコントローラ）やDMAC（DMAコントローラ）のセット（イニシャライズ）が必要なのです。そして、これこそが当初の目的であるDMAをオンにする方法だったのです。ポート31Hのビット5というのは、専用高解像度ディスプレイの行数変更に必要な処理の一部に過

ぎません。

では、なぜ桁数のほうは簡単に変更されたかというと、PC-8801ではCRTCの桁数は常に80桁に設定し、40桁へは外部回路で行っているからです。その外部回路の切り換えをするスイッチが、出力ポート30Hのビット0だったというわけです。

ということで、テキスト画面の行数変更を行いながら、CRTC／DMACをイニシャライズしてみましょう。プログラム（LIST 1.4）を実行する前に、グラフィック画面に何か表示しておいてください。

```
LINE(0,0)-(639,199),1,BF
CLEAR,&HB5FF:A=&HB700:CALL A
```

LIST 1.4

```
                        ORG     0B700H

B700 F3                 DI                   ・割込み禁止
B701 DB09       TLOOP:  IN      A,(9)
B703 07                 RLCA                 ・[ESC] が押されていればTOBSCへ
B704 3008               JR      NC,TOBSC
B706 07                 RLCA                 ・[SPACE] が押されていなければTLOOPへ
B707 38F8               JR      C,TLOOP
B709 CD10B7             CALL    CDINI        ・CRTC／DMACをイニシャライズする
B70C 18F3               JR      TLOOP        ・TLOOPへ
B70E FB         TOBSC:  EI                   ・割込み許可
B70F C9                 RET                  ・BASICへ戻る

B710 21A3B7     CDINI:  LD      HL,HICRT
B713 DB40               IN      A,(40H)
B715 E602               AND     00000010B    ・400／200ラインに合わせてHLの指定を変える
B717 2803               JR      Z,CRTOK
B719 21ABB7             LD      HL,LOCRT
B71C 3AC2E6     CRTOK:  LD      A,(0E6C2H)
B71F EE20               XOR     00100000B
B721 32C2E6             LD      (0E6C2H),A   ・テキスト画面の行数指定を変える
B724 D331               OUT     (31H),A
B726 119FB7             LD      DE,L20DT
B729 E620               AND     00100000B
B72B 2806               JR      Z,LINOK
B72D 23                 INC     HL
B72E 23                 INC     HL           ・20／25行モードでHL／DEの指定を変える
B72F 23                 INC     HL
B730 23                 INC     HL
B731 13                 INC     DE
B732 13                 INC     DE
B733 3AC0E6     LINOK:  LD      A,(0E6C0H)   ・外部回路の80／40桁モードを切り換える
B736 D330               OUT     (30H),A
B738 3AC1E6             LD      A,(0E6C1H)
B73B F608               OR      00001000B    ・CRTC同期コントロール（INACTIVE）
B73D 32C1E6             LD      (0E6C1H),A
B740 D340               OUT     (40H),A
B742 CD92B7             CALL    VWAIT        ・垂直同期期間を得る
B745 AF                 XOR     A            ・CRTCストップ
B746 D351               OUT     (51H),A
B748 3EA0               LD      A,0A0H       ・DMAストップ
B74A D368               OUT     (68H),A
B74C 3EC8               LD      A,0C8H
B74E D364               OUT     (64H),A      ・T.VRAMのトップアドレスを指定
B750 3EF3               LD      A,0F3H
B752 D364               OUT     (64H),A
B754 1A                 LD      A,(DE)
B755 D365               OUT     (65H),A
B757 13                 INC     DE           ・ターミナル・カウンタをセット
B758 1A                 LD      A,(DE)
B759 D365               OUT     (65H),A
```

```
B75B 0604               LD      B,4
B75D 7E         SDCLP:  LD      A,(HL)                ・CRTCにパラメータを送る
B75E D350               OUT     (50H),A
B760 23                 INC     HL
B761 10FA               DJNZ    SDCLP
B763 3AB9E6             LD      A,(0E6B9H)
B766 E640               AND     01000000B             ・テキスト画面のカラー／白黒をセット
B768 F613               OR      00010011B
B76A D350               OUT     (50H),A
B76C 3E43               LD      A,43H                 ・CRTCスタート
B76E D351               OUT     (51H),A
B770 3EE4               LD      A,0E4H                ・DMAスタート
B772 D368               OUT     (68H),A
B774 3E20               LD      A,20H                 ・CRT表示スタート
B776 D351               OUT     (51H),A
B778 CD98B7             CALL    VWAT2
B77B DB40       VWAT3:  IN      A,(40H)               ・水平同期期間を得る
B77D E620               AND     00100000B
B77F 20FA               JR      NZ,VWAT3
B781 3AC1E6             LD      A,(0E6C1H)
B784 E6F7               AND     11110111B             ・CRTC同期コントロール（ACTIVE）
B786 32C1E6             LD      (0E6C1H),A
B789 D340               OUT     (40H),A
B78B DB09       SPCON:  IN      A,(9)
B78D 07                 RLCA
B78E 07                 RLCA                          ・[SPACE]が押されていなければリターン
B78F 30FA               JR      NC,SPCON
B791 C9                 RET

B792 DB40       VWAIT:  IN      A,(40H)
B794 E620               AND     00100000B
B796 20FA               JR      NZ,VWAIT
B798 DB40       VWAT2:  IN      A,(40H)               ・垂直同期期間を得る
B79A E620               AND     00100000B
B79C 28FA               JR      Z,VWAT2
B79E C9                 RET

B79F 5F89       L20DT:  DB      5FH,89H               ・20行用データ
B7A1 B78B       L25DT:  DB      0B7H,8BH              ・25行用データ

B7A3 CE937338   HICRT:  DB      0CEH,93H,73H,38H      ・20行／400ライン用パラメータ
B7A7 CE986F58           DB      0CEH,98H,6FH,58H      ・25行／400ライン用パラメータ
B7AB CE9369BE   LOCRT:  DB      0CEH,93H,69H,0BEH     ・20行／200ライン用パラメータ
B7AF CE9867DE           DB      0CEH,98H,67H,0DEH     ・25行／200ライン用パラメータ
```

いかがですか。スペースキーを押すとテキスト画面の行数が変化しますが、そのたびにグラフィック画面の表示位置が上下にズレませんでしたか。すべての機種／CRTで確認したわけではありませんが、たいていはズレるはずです。これを連続して実行すると、グラフィック画面がグラグラと揺れたように見えます。そして、このときに出力ポート53Hでテキスト画面をオフにしておけば、グラフィック画面だけがワケもなく揺れたことになります。つまり、これは『グラフィック画面揺らし』というテキスト画面を使ったトリックのひとつだったのです。

CRTCやDMACの各パラメータや用法については、本書ではプログラム横のコメント以上には触れません。というのは、実際に解説するとなるとかなりのページを要する上、ほとんどゲームに利用する価値がないからです。ですから、ここでのプログラムも中身より実行結果を覚えておいてください。使いたくなったら、いつでもこのページを開けばいいのですから……。

1-4 パレットいろいろ

パレットというとBASICでも簡単に実現できるため、どうしても軽視しがちです。なにしろ画面全体が一斉に変色するわけですから、下手に操作すれば誰にでも「アッ、これはパレットでチカチカさせているんだナ!!」とわかってしまうからです。

しかし、最初の節でパレットによるプレーン分割を紹介したように、パレットにはさまざまな利用法があります。しかも、それはテキスト画面同様、イヤそれ以上にトリックのかけやすい魔法の小箱なのです。一見すると「パレットなど何もいじっていない」というようなゲームソフトが、実はパレットをいじり回していると知ったら……。おちおちゲームで遊んではかりはいられませんね。画面コントロールの基礎中の基礎テクニック、それがパレットの本当の姿です。

パレットをマシン語で操作するのは簡単です。カラー8色モードの場合、変更したいカラー番号を図1-4-1に示されるポートへダイレクトに出力すればいいのです。

パレット番号	出力ポート
0	54H
1	55H
2	56H
3	57H
4	58H
5	59H
6	5AH
7	5BH

図1-4-1　パレット番号と出力ポート

では、これを実際に実行してみることにしましょう。プログラム（LIST 1.5）は、スペースキーを押すたびにパレット番号0を次々に変化させるものです。

```
CLEAR,&HB5FF:A=&HB700:CALL A
```

LIST 1.5

```
                        ORG     0B700H

B700 F3                 DI                          ・割込み禁止
B701 DB71               IN      A,(71H)
B703 F5                 PUSH    AF                  ・メインROMセレクト
B704 3EFF               LD      A,0FFH
B706 D371               OUT     (71H),A
B708 3AD779             LD      A,(79D7H)
B70B FE33               CP      '3'
B70D 3806               JR      C,QPC88             ・PC-8801SR以降の機種はデジタルカラーモード
B70F DB32               IN      A,(32H)               にする
B711 E6DF               AND     11011111B
B713 D332               OUT     (32H),A
B715 3A34B7    QPC88:   LD      A,(PAL0C+1)
B718 C630               ADD     A,30H               ・変更したカラー番号を画面中央に表示する
B71A 3286F9             LD      (0F986H),A
```

```
B71D DB09      TLOOP:  IN     A,(9)
B71F 07                RLCA                   ・[ESC] が押されていればTOBSCへ
B720 300C              JR     NC,TOBSC
B722 07                RLCA
B723 D433B7            CALL   NC,PAL0C        ・[SPACE] が押されていればパレット0を変更する
B726 DB08              IN     A,(8)
B728 0F                RRCA                   ・[H.CLR] が押されていればDOKICをコール
B729 D44AB7            CALL   NC,DOKIC
B72C 18EF              JR     TLOOP           ・TLOOPへ
B72E F1        TOBSC:  POP    AF
B72F D371              OUT    (71H),A         ・ROMの状態を元に戻しBASICへ戻る
B731 FB                EI
B732 C9                RET

B733 3E00      PAL0C:  LD     A,0
B735 B7                OR     A               ・Aの値によって垂直同期を取る
B736 C45DB7            CALL   NZ,VWAIT
B739 3E00      P0DAT:  LD     A,0
B73B 3C                INC    A
B73C E607              AND    00000111B       ・パレット0を変化させる
B73E 323AB7            LD     (P0DAT+1),A
B741 D354              OUT    (54H),A
B743 DB09      SPOFL:  IN     A,(9)
B745 07                RLCA
B746 07                RLCA                   ・[SPACE] が押されていなければリターン
B747 30FA              JR     NC,SPOFL
B749 C9                RET

B74A 3A34B7    DOKIC:  LD     A,(PAL0C+1)
B74D EE01              XOR    1
B74F 3234B7            LD     (PAL0C+1),A
B752 C630              ADD    A,30H
B754 3286F9            LD     (0F986H),A      ・垂直同期を取るか否かのフラグを変更する
B757 DB08      HCOFL:  IN     A,(8)
B759 0F                RRCA
B75A 30FB              JR     NC,HCOFL
B75C C9                RET

B75D DB40      VWAIT:  IN     A,(40H)
B75F E620              AND    00100000B
B761 20FA              JR     NZ,VWAIT
B763 DB40      VWAT2:  IN     A,(40H)         ・垂直同期期間を得る
B765 E620              AND    00100000B
B767 28FA              JR     Z,VWAT2
B769 C9                RET
```

画面中央付近に '0' と表示され、スペースキーと連動してパレット0が変わります。ここでスペースキーを速射砲のように連打してみてください。パレットの変化に伴い、画面を寸断するようなチラツキが気になってきます。これは、パソコンのディスプレイが図1-4-2のような順で1/60秒毎に表示を繰り返しているため、表示の途中でパレットを変更すると瞬間的に画面の上下で色が変わってしまうからです。

これを避けるためには、表示が完了して次の表示が始まるまでの間（走査線が上にもどる垂直帰線期間内）にパレット変更をすればいいのです。今度は**H.CLR**キーを押して同じようにしてみてください。画面中央には違いを示すため '1' と表示され、パレット変化の際のチラツキはなくなります。本来ならば、グラフィックVRAMをアクセスするときにもこのように同期を取るべきなのですが、速度的な問題から88では特殊なケース以外は行いません。

なお、本プログラムでは2度目の垂直帰線期間をポーリング（サーチ）すること

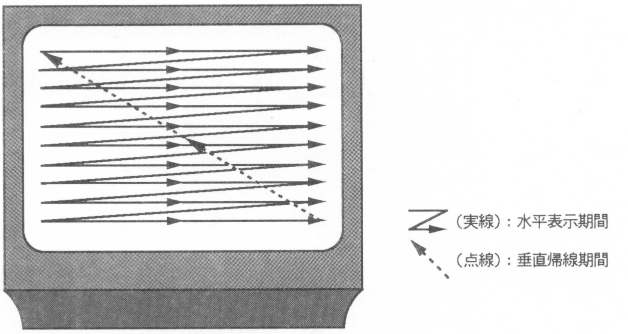


図1-4-2　ハードウェアから見た画面表示の実体

で、正確な垂直帰線のスタートを得ていましたが、実は1／60秒毎のVRTC割込みとはこの垂直帰線のスタートで発生する割込みのことなのです。したがって、VRTC割込みでパレットを変化させる場合は、このような気を使う必要はありません。

このプログラムでもう1つ注目したいのは、機種の判別方法についてです。PC-8801シリーズはPC-8801SRを境にしてハードウェアの機能が大幅にアップし、プログラムを共通化するためには機種の判別が欠かせないのです。詳しい変貌は『マシン語サウンド・プログラミング』で解説していますので、ここではBASIC ROM内の79D7Hを見る方法と理由についてだけ述べます。

BASICからCALL文によりマシン語プログラムへ入ると、PC-8801SR以降の機種においてはメインROMが選択されておらず、そのままでは79D7Hの値を正しく読めません。これはCALL文を処理するプログラムがPC-8801SR以降ROM化（それまではDISK BASICでサポート）され、別の場所（4th ROM）に置かれたためです。そこで、79D7Hの値を見たりメインROM内のルーチンを利用する際には、プログラムの前でメインROMを選択し、プログラム終了後にそれを元にもどすという作業が必要になったのです。

PC-8801SR以降（V2モード）では、カラーパレットは標準でアナログ512色モードに設定されており、今回のように8色にするためにはモードを変更しなければなりません。このこと自体は、ポート32Hのビット5の値を反転すればいいのですが、このポート32HはPC-8801mkⅡ以前の機種ではデコードされておらず、他のポートへ影響を及ぼす危険性があるのです。そのため、面倒でも機種の判別をしてからそれを実行しているわけです。つまらないことのようですが、正体不明のバグに出会わないよう、このあたりにも細心の注意を払わなければなりません。

次に、アナログ512色モードでのパレット変更を行ってみましょう。アナログ512色といっても、結局はそこから8色を選ぶわけですから、使用するポートは図1-4-1と同じです。ただし、カラー番号として0～511を直接指定するのではなく、B／R／Gそれぞれの輝度（8レベル：0～7）で指定します。8レベル（0～7）を表現するのに3ビット必要ですから、当然1バイト（8ビット）では全プレーンを指定できません。そこで、図1-4-3のようにビット6の違いでB／R面とG面とを区別し、

2度に分けて指定するようになっています。

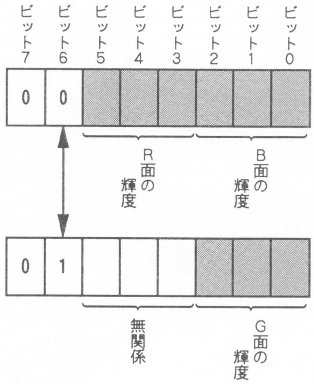


図1-4-3　アナログカラーの指定法

プログラム（LIST 1.6）は、先ほどと同じような感覚でパレット0を輝度別に変化させるものです。それぞれの輝度は[f･1]～[f･3]を押しながら[4]～[6]キーで連続して調整することができます。また、垂直帰線の同期は[H.CLR]でオン／オフができます。

CLEAR,&HB5FF:A=&HB700:CLS:CALL A

LIST 1.6

```
                    ORG     0B700H

B700 F3             DI                      ・割込み禁止
B701 DB71           IN      A,(71H)         ┐
B703 F5             PUSH    AF              │
B704 3EFF           LD      A,0FFH          ├・メインROMセレクト
B706 D371           OUT     (71H),A         ┘
B708 3AD779         LD      A,(79D7H)       ┐
B70B FE33           CP      '3'             ├・PC-8801SR以前の機種はTOBSCへ
B70D 385E           JR      C,TOBSC         ┘
B70F DB32           IN      A,(32H)         ┐
B711 F620           OR      00100000B       ├・アナログカラモードにする
B713 D332           OUT     (32H),A         ┘

B715 21C8F3         LD      HL,0F3C8H       ┐
B718 117700         LD      DE,120-1        │
B71B 3656           LD      (HL),'V'        │
B71D 23             INC     HL              │
B71E 363D           LD      (HL),'='        │
B720 19             ADD     HL,DE           │
B721 3642           LD      (HL),'B'        │
B723 23             INC     HL              │
B724 363D           LD      (HL),'='        │
B726 19             ADD     HL,DE           │
B727 3652           LD      (HL),'R'        ├・テキスト画面の初期設定
B729 23             INC     HL              │
B72A 363D           LD      (HL),'='        │
B72C 19             ADD     HL,DE           │
B72D 3647           LD      (HL),'G'        │
B72F 23             INC     HL              │
B730 363D           LD      (HL),'='        │
B732 3A8DB7         LD      A,(PAL0C+1)     │
B735 C630           ADD     A,30H           │
B737 32CAF3         LD      (0F3C8H+2),A    │
B73A CDA5B7         CALL    DTDIP           ┘
```

```
B73D DB08       TLOOP:  IN      A,(8)           ・[H.CLR] が押されていればDOKICをコール
B73F 0F                 RRCA
B740 D4BEB7             CALL    NC,DOKIC
B743 DB09               IN      A,(9)           ・[ESC] が押されていればTOBSCへ
B745 07                 RLCA
B746 3025               JR      NC,TOBSC
B748 5F                 LD      E,A             ・[f.1] が押されていればB面の輝度値をチェック
B749 1600               LD      D,0
B74B 2172B7             LD      HL,ANABB
B74E CB53               BIT     1+1,E
B750 CC75B7             CALL    Z, ACHAN
B753 23                 INC     HL              ・[f.2] が押されていればR面の輝度値をチェック
B754 CB5B               BIT     2+1,E
B756 CC75B7             CALL    Z,ACHAN
B759 23                 INC     HL              ・[f.3] が押されていればG面の輝度値をチェック
B75A CB63               BIT     3+1,E
B75C CC75B7             CALL    Z,ACHAN
B75F 15                 DEC     D               ・D＝1（輝度値変化あり）ならパレットを変更する
B760 CC8CB7             CALL    Z,PAL0C
B763 019600             LD      BC,150          ・ウェイトを取ってTLOOPへ
B766 10FE       TWAIT:  DJNZ    $
B768 0D                 DEC     C
B769 20FB               JR      NZ,TWAIT
B76B 18D0               JR      TLOOP
B76D F1         TOBSC:  POP     AF              ・ROMの状態を元に戻しBASICへ戻る
B76E D371               OUT     (71H),A
B770 FB                 EI
B771 C9                 RET

B772 00         ANABB:  DB      0               ・B面の輝度値
B773 00         ANARR:  DB      0               ・R面の輝度値
B774 00         ANAGG:  DB      0               ・G面の輝度値

B775 4E         ACHAN:  LD      C,(HL)          ・[4] ～ [6] が押されていれば輝度値を変更する
B776 DB00               IN      A,(0)
B778 CB67               BIT     4,A
B77A 2001               JR      NZ,NOTK4
B77C 0D                 DEC     C
B77D CB77       NOTK4:  BIT     6,A
B77F 2001               JR      NZ,NOTK6
B781 0C                 INC     C
B782 79         NOTK6:  LD      A,C
B783 FE08               CP      8
B785 D0                 RET     NC
B786 BE                 CP      (HL)
B787 C8                 RET     Z
B788 77                 LD      (HL),A
B789 1601               LD      D,1
B78B C9                 RET

B78C 3E00       PAL0C:  LD      A,0             ・Aの値によって垂直同期を取る
B78E B7                 OR      A
B78F C4D1B7             CALL    NZ,VWAIT
B792 2172B7             LD      HL,ANABB        ・アナログパレットの変更
B795 3A73B7             LD      A,(ANARR)
B798 07                 RLCA
B799 07                 RLCA
B79A 07                 RLCA
B79B B6                 OR      (HL)
B79C D354               OUT     (54H),A
B79E 3A74B7             LD      A,(ANAGG)
B7A1 F640               OR      01000000B
B7A3 D354               OUT     (54H),A
B7A5 3A72B7     DTDIP:  LD      A,(ANABB)       ・各面の輝度値を表示する
B7A8 C630               ADD     A,30H
B7AA 3242F4             LD      (0F3C8H+122),A
B7AD 3A73B7             LD      A,(ANARR)
B7B0 C630               ADD     A,30H
B7B2 32BAF4             LD      (0F3C8H+242),A
B7B5 3A74B7             LD      A,(ANAGG)
B7B8 C630               ADD     A,30H
B7BA 3232F5             LD      (0F3C8H+362),A
B7BD C9                 RET
```

```
B7BE 3A8DB7   DOKIC:  LD    A,(PAL0C+1)
B7C1 EE01             XOR   1
B7C3 328DB7           LD    (PAL0C+1),A
B7C6 C630             ADD   A,30H
B7C8 32CAF3           LD    (0F3C8H+2),A    ・垂直同期を取るか否かのフラグを変更する
B7CB DB08     HCOFL:  IN    A,(8)
B7CD 0F               RRCA
B7CE 30FB             JR    NC,HCOFL
B7D0 C9               RET

B7D1 DB40     VWAIT:  IN    A,(40H)
B7D3 E620             AND   00100000B
B7D5 20FA             JR    NZ,VWAIT
B7D7 DB40     VWAT2:  IN    A,(40H)         ・垂直同期期間を得る
B7D9 E620             AND   00100000B
B7DB 28FA             JR    Z,VWAT2
B7DD C9               RET
```

画面上部には、垂直同期オン／オフの状態（１でオン）とパレット０の各プレーンの輝度が表示されます。こうして連続して輝度を変化させると、垂直同期を取る重要性が改めて感じられるかもしれません。

V=0…垂直同期の状態
B=0…B面の輝度レベル
R=0…R面の輝度レベル
G=0…G面の輝度レベル

アナログカラーはPC-8801SR以降で追加された機能ですから、PC-8801mkⅡ以前の機種では当然対応できません。無理に実行すれば出力した下位３ビットのカラーに変化しますが、このような場合は「PC-8801SR以降専用」としてしまうのが普通です。このプログラム（LIST 1.6）でも、何もせずにBASICへもどるようにしています。

では、PC-8801SR以降であってもディスプレイがアナログに対応していない場合はどうでしょうか。プログラムでいちいち区別するのでは面倒です。しかし、実はこれは心配することはありません。ディスプレイがデジタルの場合は、輝度レベルが少し（０以外）でもあれば、すべて輝度７（最大）で発色するようになっており、自動的に中間輝度が省かれるのです。どうしても特殊な中間色で固定したい場合は、「要アナログ・ディスプレイ」としなければなりませんが、実際のゲームでは８色でもおかしくない色に設定するか、中間色を単にパレットのなめらかな変化に利用して、デジタル・ディスプレイでも支障のないようにしておくのが一般的です。こうすれば、アナログ・ディスプレイの方はより以上に美しい画面で、デジタル・ディスプレイの方はそれなりの画面でゲームを楽しむことができるからです。

パレットの具体的な応用例については別の章で示しますが、プレーンの状態とカラー番号との関係（図1-1-2参照）を熟知しておくことが最大のキーポイントです。パレットをいろいろと変化させながら、カラー番号（0〜7）を聞いただけでプレーンの状態（オンかオフか）がすぐわかるよう慣れ親しんでください。

1-5 メモリがいっぱい

最近では、テープ版のゲームなどすっかり姿を消してしまいましたが、ディスクがここまで普及すればそれも止むを得ないことでしょう。それに、そのお陰で巨大なスケールのゲームが手軽に楽しめるようになったのですから、困ることなど何もないはずです。制作する側の苦労が増えたこと以外は……。

しかし、実際にはこれはプレイヤーにも関係があることです。ディスクという膨大なメモリ空間を扱うようになると、オンメモリで扱える能力もそれなりにアップしてくれないとイライラだけがアップしてしまいます。まるで'おちょこ'でご飯を食べさせられるような気分かもしれません。もはやメモリ不足はプログラマーにとってもプレイヤーにとっても最大の悩みといえるでしょう。これはテープ版のソフトを制作していたころには考えられなかったことです。

そこで、どこかに空いているメモリはないかと捜すわけですが、PC-8801mkⅡまではグラフィックVRAMをパレットで隠して使う程度しかできませんでした。どこを捜してもナイものはナイのですから。しかし、PC-8801SR以降の機種にはほんの4Kバイトですが、標準で使えるRAMが増えたのです。これがどれだけ有難いか、メモリ不足で悩んだことがない人にはわからないことでしょう。それは、乾ききった砂漠で飲むコップ一杯の水に相当するほど貴重な存在なのです。

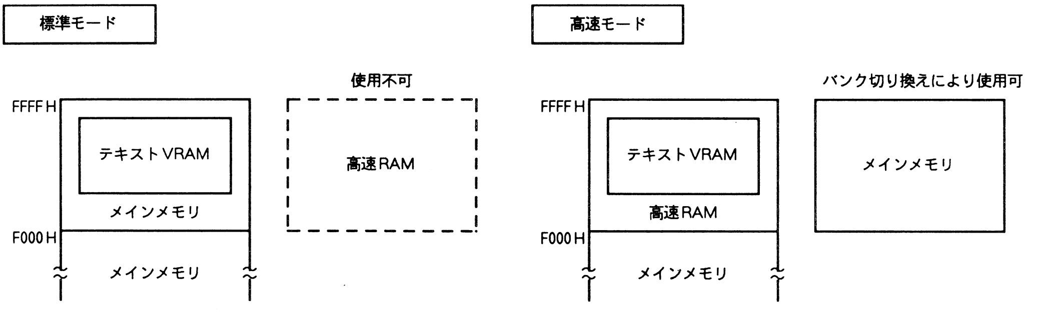


図1-5-1 PC-8801SR以降のテキストVRAM

そのメモリは、ちょうどテキストVRAMエリアの裏側を覆う形でF000H～FFFFHに存在しており、バンク切り換えにより使用することができます。「1-3 テキスト画面の豆知識」で『PC-8801SR以降においてはDMAによるCPUの一時停止がなくなった』と書いてあったことを思いだしてください。実は、これはテキストVRAMをメインメモリから高速RAMへ移した結果実現したものなのです。標準スピードモードでは、テキストVRAMはメインメモリ上にありますが、高速モード（V1H,V2）下ではF000H～FFFFHが高速RAMとなり（図1-5-1）、メインメモリは切り離されています。

この未使用のメインメモリを必要に応じ高速 RAMと切り換えて使用すれば、

トータルして68Kのメモリが自由に使えるわけです。もちろん、フルに68Kを使うためにはBASICのシステムに依存しないだけのプログラミング知識が要求されますが、たとえBASICと共存していてもグラフィックVRAMのバンク切り換えと同じ要領で管理するだけで、+4Kバイトのメモリ空間を得ることができます。

バンク切り換えの方法は、プログラム（LIST 1.7）で示されているようにポート32Hのビット4を1にして出力するだけです。

```
CLEAR,&HB5FF:A=&HB700:CLS:CALL A
```

LIST 1.7

```
                        ORG     0B700H

B700 F3                 DI                       ・割込み禁止
B701 DB71               IN      A,(71H)
B703 F5                 PUSH    AF
B704 3EFF               LD      A,0FFH           ・メインROMセレクト
B706 D371               OUT     (71H),A
B708 3AD779             LD      A,(79D7H)
B70B FE33               CP      '3'              ・PC-8801SR以前の機種はTOBSCへ
B70D 3837               JR      C,TOBSC
B70F DB32               IN      A,(32H)
B711 F5                 PUSH    AF
B712 F610               OR      00010000B        ・メインメモリにバンク切り換え
B714 D332               OUT     (32H),A
B716 214BB7             LD      HL,MDATA
B719 1196F8             LD      DE,0F896H
B71C 010600             LD      BC,6             ・文字データをメインメモリに送る
B71F EDB0               LDIR
B721 F1                 POP     AF               ・高速RAMに戻す
B722 D332               OUT     (32H),A
B724 0606               LD      B,6
B726 2196F8             LD      HL,0F896H
B729 DB09       TLOOP:  IN      A,(9)
B72B 07                 RLCA
B72C 3018               JR      NC,TOBSC
B72E 07                 RLCA
B72F 38F8               JR      C,TLOOP
B731 DB32               IN      A,(32H)
B733 F5                 PUSH    AF
B734 F610               OR      00010000B        ・[SPACE]が押されるたびにメインメモリから高速
B736 D332               OUT     (32H),A           RAMへデータを転送する
B738 5E                 LD      E,(HL)
B739 23                 INC     HL
B73A F1                 POP     AF
B73B D332               OUT     (32H),A
B73D 73                 LD      (HL),E
B73E DB09       SOFLP:  IN      A,(9)
B740 07                 RLCA
B741 07                 RLCA
B742 30FA               JR      NC,SOFLP
B744 10E3               DJNZ    TLOOP
B746 F1         TOBSC:  POP     AF
B747 D371               OUT     (71H),A
B749 FB                 EI                       ・ROMの状態を元に戻しBASICへ戻る
B74A C9                 RET

B74B D2B2DDD2   MDATA:  DB      'ﾒｲﾝﾒﾓﾘ'         ・文字データ
B74F D3D8
```

このプログラムでは、まずメインメモリにバンク切り換えをしてF896H番地からへ‘メインメモリ’という文字データを送ります。次に、スペースキーが押されるたびにバンク切り換えをしながらそれを高速RAMの同じアドレス、つまりテキ

ストVRAMの中央付近に転送し＋4Kバイトのメモリを確認するものです。テキスト画面に表示される文字を見ながら、増えたメモリの活用法でも考えてください。

しかし、現実にはこの程度でメモリ不足が解消されるわけはありません。そこで、ますます機種は限定されますが、Mシリーズ*（MR, MH, MA, MA 2：1989年7月現在）専用とするならば､メモリはさらに128Kバイトを使うことができます。

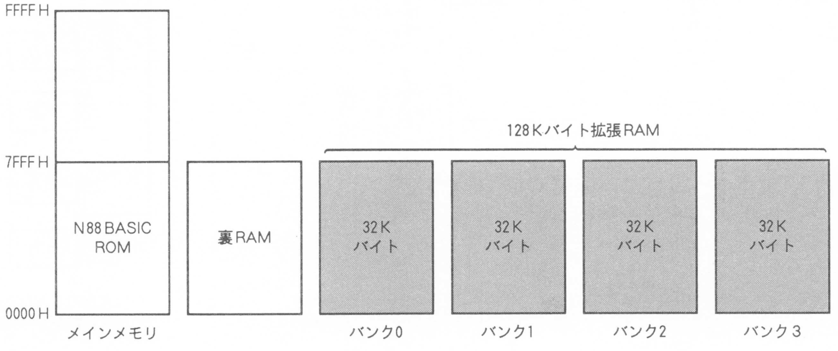


図1-5-2　拡張RAM

図1-5-2に示されている拡張RAMは、4バンク合わせてカードと呼ばれ、オプションでさらに3枚計4枚のカードを装備することができます。これらは0～3のカード番号で区別され、標準で装備されているのはカード番号0となっています。この拡張RAMを使用する際には、まずカード番号と拡張RAMのバンク番号を選択します。そして、そのバンクでのリード／ライトを指定します。ここで初めて拡張RAMの利用が可能となります。したがって、E3Hによる拡張RAM指定だけではバンク切り換えとならず、E2Hでのリード／ライトの指定が実際のバンク切り換えということです。図1-5-3が、指定のためのポート内容です。

一応、どちらのポートも入出力が可能となっていますが、拡張ボードを2枚以上実装した場合にはうまく読めないことがあるようなので、実際には自分で利用状況をワークエリアにキープしておいたほうがいいでしょう。参考までに、拡張RAM（カード0）のリード／ライトに関する用法と内容を示しておきます。

(1) リードのみを行う場合

ポートE3Hに000000XXB（XXでバンクを指定）を出力

ポートE2Hに00000001Bを出力

※以降、アドレス0～7FFFHに対するリードは拡張RAMに、ライトは裏RAMに対して行われる。

メモリの増設：

Mシリーズ以外の機種でも、マルチボード（PC-8801-22等）を増設すればメモリ的にはMシリーズと同等とすることができる。ただし、ボードによっては4MHzで動作させなければならないものもあるので、使用の際にはカタログなどでよく確認すること。

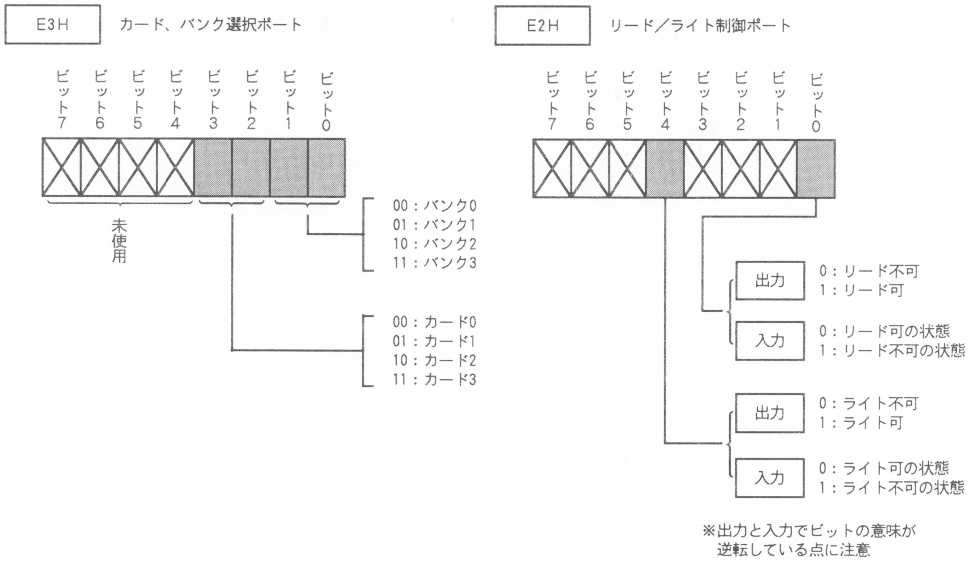


図1-5-3　拡張RAM利用のためのポート（入出力可能）

(2)ライトのみを行う場合

ポートE3Hに000000XXB（XXでバンクを指定）を出力

ポートE2Hに00010000Bを出力

※以降、アドレス0〜7FFFHに対するリードはメインメモリ（N88 BASIC ROM）に、ライトは拡張RAMに対して行われる。

(3)リード／ライトを行う場合

ポートE3Hに000000XXB（XXでバンクを指定）を出力

ポートE2Hに00010001Bを出力

※以降、アドレス0〜7FFFHに対するリード／ライトはいずれも拡張RAMに対して行われる。

なお、E2Hポートによる拡張RAMの制御は、ポート31Hでの拡張ROM（4th ROM）のセレクトや、ポート71Hでのメインメモリ／裏RAMの選択状態にかかわらず、上述した指定のメモリに対して行われます。

これは、もう砂漠でコップ一杯の水どころではなく、広大な湖に出会ったようなものです。今後は、Mシリーズ専用のソフトも増えてくるかもしれません。ところで、なぜグラフィックの基礎の章にこのようなメモリの節があるのか、疑問に思われた方も多いことでしょう。この理由は単純です。それは、グラフィック・データが最も大量にメモリを消費するからです。グラフィックを追求すれば、メモリに関する知識は基礎として避けて通れない道なのです。

もしも最初からこれくらいのメモリがあったら……。それでも、もっとメモリがほしいと嘆いていたでしょうか。食欲は満腹になれば終わりですが、金欲とメモリ欲には際限がなさそうで、非常にコワイ気がします。

第2章　拡張グラフィックスのすべて

『斬り捨て御免……』

こんな言葉は江戸時代のことかと思っていたら、なんとこの近代社会においても『足切り』などという妙なモノに姿を変えて残っているようです。時代は違えど、切られる側はたまったものではありません。とはいえ、現代は自由競争の社会。勝者がいれば必ず敗者がでる仕組みなのであります。

無差別に『差別』をするのは問題ですが、ヤヤコシイことに『差別』と『区別』は区別をしなければなりません。例えば、ホテルや旅館の大浴場。男風呂と女風呂とを分けるのは区別でも、男風呂が大きくて女風呂が小さいと「これは差別だ!!」と問題になるわけです。ところで……。

『PC-8801SR以降専用』というのは差別か区別か……。私には区別ができませんが、少なくともそれまでの機種（PC-8801mkII以前）との間に違いがあるのは事実です。その事実を正確に認識することは、PC-8801mkII以前のユーザーにとっても決してマイナスではないと思うのですが……。この章は、『PC-8801SR以降専用』です。



2-1 ALUとは

よく言葉が一人歩きするといいますが、ALUもPC-8801SR以降になって急激に脚光を浴びた名称です。最近のゲームソフトがPC-8801SR以降となった理由にも、このALUはたびたび登場してきます。しかし、実はALU（Arithmetic and Logic Unit＝算術論理演算回路）そのものはPC-8801mkⅡ以前にも立派に存在しているのです。それどころか、どんなコンピュータにもある当然の機能なのです。

では、ALUとは何かということになりますが、これは文字通り数値を演算する回路のことで、例えばAレジスタを＋5するという場合、実際にその計算をしているのがALUなのです。一般にはこういった計算はCPUが行っていると理解すれば十分なのですが、正しくはCPU内部のALUが計算を実行し、その結果をレジスタやフラグに反映しているというわけです。

こう書いてしまうと、それでは一人歩きしたほうのALUはどうなってしまったのか不安になるかもしれません。ワケもなく知名度だけがアップしたのでしょうか。そんなはずはありませんね。これは、グラフィックVRAMのデータと自動的に論理演算を行うという新機能のことなのです。つまり、従来のALUに新たにグラフィック専用の演算回路を追加したものが、本当のPC-8801SR以降のALUの姿なのです。しかし、従来のALUが存在自体を気にする必要がなかったのに対し、この新しい機能は使用宣言をしなければ使うことができません。このため、意識しないで使っているほうのALUは無視され、ALUといえばグラフィック専用の演算回路を指すようになったのです。

ところで、実際にPC-8801SR以降で拡張されたグラフィックス機能は、このALUだけではありません。それ以外にもいろいろな機能が追加されています。ALUはそういった拡張グラフィックス機能の一部に過ぎないのですが、拡張グラフィックスという新鮮味のない言葉に比べ、ALUという名称は響きもナウいしシンプルです。その結果、ALUは拡張グラフィックス機能の代名詞として完全に独立した意味を持ってしまったのです（ただし、あくまでも俗称です）。

本来、言葉というものは意思伝達の手段ですから、どう表現しようと機能そのものに影響を与えることはありません。ましてや、ALUで意思の疎通ができるならば、わざわざ拡張グラフィックスなどとする必要はないかもしれません。しかし、本書ではその事実は事実として認めた上で、キチンとした区別をつけて記述をしています。これは、俗称を修正しようという目的ではなく、ALUの真の姿を確認してもらいたいからです。

さて、新たに拡張されたグラフィックス処理機能ですが、グラフィックVRAMをアクセスするという点では従来のバンク切り換えと変わりありません。したがって、次のようなことに注意するのは当然です。

・プログラムやスタックポインタはC000H以前に置く
・割込みを禁止（または自分で管理）する

また、この拡張グラフィックス機能は従来のバンク切り換えモード（以下、独立アクセスモードと呼ぶ）と同じ次元では使用できません。最初に拡張グラフィックス機能を使用可能なモード（以下、拡張アクセスモードと呼ぶ）へスイッチを切り換えることで使用可能となります。これにはポート32Hのビット6を1にして出力します。

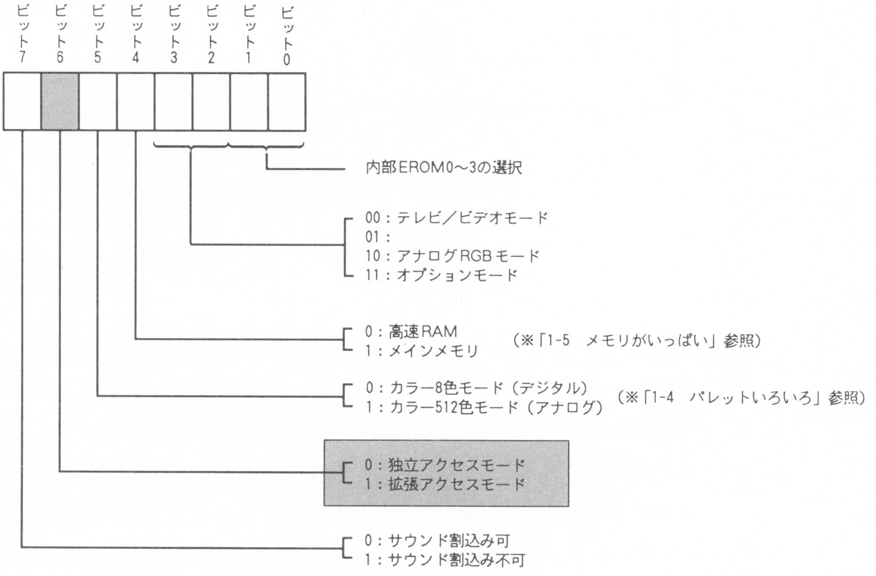


図2-1-1　ポート32Hの内容（入出力可）

拡張グラフィックス機能を実際に利用するためには、ここで独立アクセスモードでのバンク切り換え（OUT 5CH～5EH）に相当する手順が必要です。つまり、メインメモリからグラフィックVRAMをアクセスするモードへの切り換えです。そして、このときに選択できる機能は独立アクセスモードと違って多岐にわたります。この節では全機能を一覧できるようにしていますが、ここですべてを覚えることはありません。必要に応じて、その都度確認すればいいのです。また、それをサポートするために次節からは具体的な利用法について示していきます。

まず、拡張アクセスモードでグラフィックVRAMを操作するためのI／Oポートですが、これには34Hと35Hの2つがあります。このうち、メインになるのは35Hのほうです。図2-1-2を見てください。

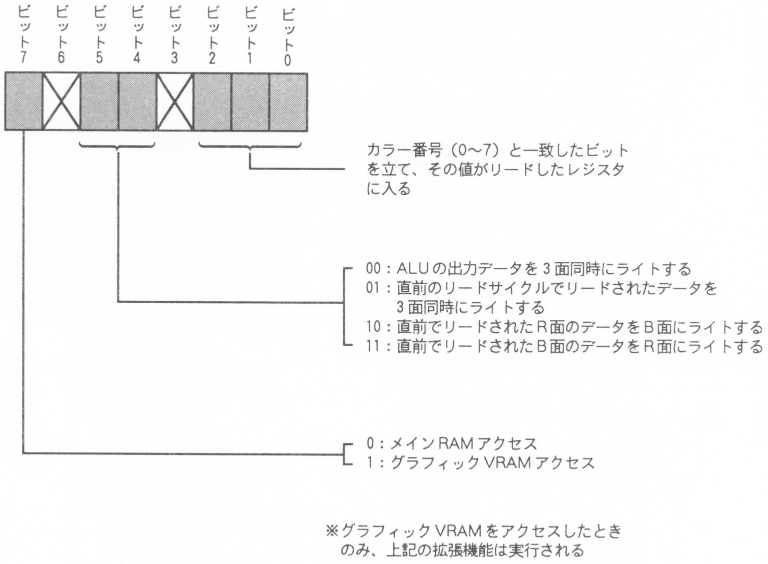


図2-1-2　出力ポート35Hの内容

ビット7の値により、C000H以降へのアクセスがグラフィックVRAMになるかメインRAMになるかが決まります。したがって、グラフィックVRAMアクセス時にはこのビットを1にし、ビット4〜5で機能を選択します。ビット0〜2はそこで選択された機能とは無関係の働きをするもので、グラフィックVRAMの内容をリードすると自動的にそのカラーのビットだけを1（それ以外は0）にし、リードしたレジスタに入れるというものです。これはグラフィックVRAMからドット単位でカラー判定をするときに便利な機能です。

ビット4〜5の役割は、図2-1-2の説明ではまだわかりにくいかもしれません。とりあえずは『00：ALUの出力データを3面同時にライトする』という部分だけ注目してください。それ以外のものについては、次節以降の応用例で解説します。

やっとALUという言葉が登場しましたが、ここでいうALUの出力データとは「数値（あるいはレジスタ値）とグラフィックVRAMのデータとの論理演算をしたもの」という意味です。そして、これを数値（あるいはレジスタ値）をグラフィックVRAMにストアするだけで実現できるのがALUの新機能なのです。しかも、グラフィックVRAMのデータとの論理演算は、B／R／Gそれぞれの面に対し独立して指定することができます。その指定を行う出力ポートが34Hです。

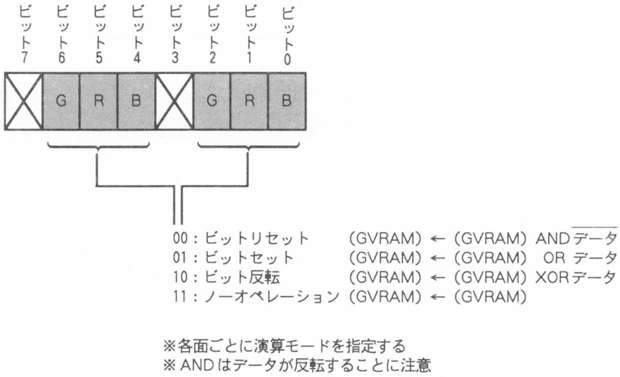


図2-1-3　出力ポート34Hの内容

図2-1-3で示されているように、各面との論理演算の種類は2ビットを使って指定します。面ごとにビットが離れているので、慣れるまではセットしにくいかもしれません。しかし、実用上はそのほうが便利なことが多いのです。例えば、第1章に点を打つプログラム（LIST 1.1）がありましたが、同じことをALUを使って実現してみましょう。

LIST 2.1

```
                    ORG     0B700H

B700 F3             DI                          ・割込み禁止
B701 2100C0         LD      HL,0C000H           ・HL = G.VRAM
B704 DB32           IN      A,(32H)             ┐
B706 CBF7           SET     6,A                 ├・拡張アクセスモード
B708 D332           OUT     (32H),A             ┘
B70A 3E80           LD      A,10000000B         ┐・ALU3面同時ライトモードにし、メモリはG.
B70C D335           OUT     (35H),A             ┘  VRAMをセレクトする
B70E 3E05           LD      A,00000101B         ┐・B／G面＝ドットセット、R面＝ドットリセット
B710 D334           OUT     (34H),A             ┘
B712 3680           LD      (HL),10000000B      ・点を打つ
B714 AF             XOR     A                   ┐・メインメモリをセレクト
B715 D335           OUT     (35H),A             ┘
B717 DB32           IN      A,(32H)             ┐
B719 CBB7           RES     6,A                 ├・独立アクセスモード
B71B D332           OUT     (32H),A             ┘
B71D FB             EI                          ・割込み許可
B71E C9             RET                         ・BASICへ戻る
```

プログラム（LIST 2.1）からALUのモード設定をしている部分をカットすると、データをグラフィックVRAMに1回ストアしているだけです。これだけで、各面に対する『ANDを取ってORを取る』という作業も完了なのです。そして、このとき34Hへ出力する値がカラー番号そのままということがポイントです。ALUによる論理演算の内容はどのようなものだったのか、図2-1-3と照らし合わせながらよく確認してください。

また、ALUの設定をいろいろと変化させて研究するのも役に立つことです。もしかすると、自力でALUの欠点を発見できるかもしれません……。

2-2 重ね合わせ

PC-8801シリーズの画面構成（図1-1-1）から判断すると、グラフィック画面どうしの重ね合わせ処理など不可能なように見えます。しかし、それが可能であることはすでに多くのゲームによって証明されていますし、おそらく不可能と信じている人など1人もいないでしょう。それどころか、図1-1-1以外に特殊なキャラクタ表示機能を持っているはずだと信じている人さえいるかもしれません。

残念なことに、たとえPC-8801SR以降であっても、グラフィックスに関してのハードウェア的な能力はこれまでに説明してきたことがすべてです。これ以外のことは、すべてプログラム・テクニックによってカバーされているのです。もちろん、最初から多様なテクニックが存在していたわけではありません。ハードウェアで用意されていない機能は、プログラマーが夢を追い求めて実現したものばかりです。つまり、PC-8801シリーズ（SR以降）というのは「テクニックを要求するが、テクニックの酷使にも耐えられる」ように設計された機種なのです。

画面処理テクニックへの第一歩、それはキャラクタの重ね合わせです。すでに基本的な原理はドット表示の際に示されています。これをキャラクタに拡大したのが次のプログラム（LIST 2.2）です。まずは、実行してみてください。

```
CLEAR,&HB5FF:A=&HB700:CALL A
```

LIST 2.2

```
                         ORG     0B700H

B700 F3                  DI                       ・割込み禁止
B701 3E99                LD      A,10011001B     ┐
B703 D35C                OUT     (5CH),A         │
B705 CD2FB7              CALL    BACKD           │
B708 3E33                LD      A,00110011B     │
B70A D35D                OUT     (5DH),A         ├・背景作成
B70C CD2FB7              CALL    BACKD           │
B70F 3ECC                LD      A,11001100B     │
B711 D35E                OUT     (5EH),A         │
B713 CD2FB7              CALL    BACKD           │
B716 D35F                OUT     (5FH),A         ┘
B718 DD21A3B7            LD      IX,BMENS        ┐
B71C D35C                OUT     (5CH),A         │
B71E CD42B7              CALL    DICHR           │
B721 D35D                OUT     (5DH),A         ├・おばけのピーヨ表示
B723 CD42B7              CALL    DICHR           │
B726 D35E                OUT     (5EH),A         │
B728 CD42B7              CALL    DICHR           │
B72B D35F                OUT     (5FH),A         ┘
B72D FB                  EI                       ・割込み許可
B72E C9                  RET                      ・BASIC へ戻る

B72F 2100C0     BACKD:   LD      HL,0C000H       ┐
B732 0E64                LD      C,100           │
B734 06A0       BACL0:   LD      B,80*2          │
B736 77         BACL1:   LD      (HL),A          │
B737 07                  RLCA                    ├・G.VRAM全体をAレジスタで埋める（2ライン毎に
B738 07                  RLCA                    │  2ローテート）
B739 23                  INC     HL              │
B73A 10FA                DJNZ    BACL1           │
B73C 07                  RLCA                    │
```

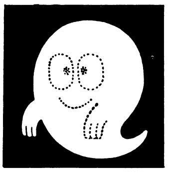
```
B73D 07                 RLCA
B73E 0D                 DEC     C
B73F 20F3               JR      NZ,BACL0
B741 C9                 RET

B742 1163B7    DICHR:   LD      DE,TMEN1
B745 21D0DD             LD      HL,0DDD0H
B748 0E10               LD      C,16
B74A 0604      DICL0:   LD      B,4
B74C 1A        DICL1:   LD      A,(DE)
B74D A6                 AND     (HL)
B74E DDB600             OR      (IX+0)
B751 77                 LD      (HL),A
B752 23                 INC     HL
B753 13                 INC     DE
B754 DD23               INC     IX        ・透明データで画面をくり抜いて (AND) からキャラクタ
B756 10F4               DJNZ    DICL1       を表示 (OR)
B758 3E4C               LD      A,80-4
B75A 85                 ADD     A,L
B75B 3001               JR      NC,AHL80
B75D 24                 INC     H
B75E 6F        AHL80:   LD      L,A
B75F 0D                 DEC     C
B760 20E8               JR      NZ,DICL0
B762 C9                 RET

B763 FFE003FF  TMEN1:   DB      0FFH,0E0H,3,0FFH,0FFH,0,0,7FH ・くり抜き用透明データ
B767 FF00007F
B76B FC00001F           DB      0FCH,0,0,1FH,0F8H,0,0,0FH
B76F F800000F
B773 F0000007           DB      0F0H,0,0,7,0E0H,0,0,3
B777 E0000003
B77B E0000001           DB      0E0H,0,0,1,0E0H,0,0,1
B77F E0000001
B783 E0000001           DB      0E0H,0,0,1,0C0H,0,0,1
B787 C0000001
B78B 80000003           DB      80H,0,0,3,0,0,0,5
B78F 00000005
B793 00000000           DB      0,0,0,0,83H,0,0,1
B797 83000001
B79B FFE00003           DB      0FFH,0E0H,0,3,0FFH,0FEH,0,0FH
B79F FFFE000F
B7A3 00000000  BMENS:   DB      0,0,0,0,0,1FH,0FCH,0          ・B面のデータ
B7A7 001FFC00
B7AB 00FFFF80           DB      0,0FFH,0FFH,80H,3,0FFH,0FFH,0E0H
B7AF 03FFFFE0
B7B3 070F0FF0           DB      7,0FH,0FH,0F0H,0EH,6,7,0F8H
B7B7 0E0607F8
B7BB 0E0607FC           DB      0EH,6,7,0FCH,0FH,0FH,0FH,0FCH
B7BF 0F0F0FFC
B7C3 0FFFFFFC           DB      0FH,0FFH,0FFH,0FCH,1FH,0FFH,0FFH,0FCH
B7C7 1FFFFFFC
B7CB 2B9F3278           DB      2BH,09FH,32H,78H,55H,0E0H,0E4H,0F0H
B7CF 55E0E4F0
B7D3 00FFE0E2           DB      0,0FFH,0E0H,0E2H,0,1FH,0FFH,8CH
B7D7 001FFF8C
B7DB 0001FFF0           DB      0,1,0FFH,0F0H,0,0,0,0
B7DF 00000000
B7E3 00000000  RMENS:   DB      0,0,0,0,0,6,0D0H,0            ・R面のデータ
B7E7 0006D000
B7EB 003FFA00           DB      0,3FH,0FAH,0,1,0FH,0EH,80H
B7EF 010F0E80
B7F3 02F6F740           DB      2,0F6H,0F7H,40H,5,0F9H,0FBH,0A0H
B7F7 05F9FBA0
B7FB 05C93BD0           DB      5,0C9H,3BH,0D0H,2,0F6H,0F7H,0A0H
B7FF 02F6F7A0
B803 070F0FD0           DB      7,0FH,0FH,0D0H,1BH,0FFH,0FFH,0A0H
B807 1BFFFFA0
B80B 399F3720           DB      39H,09FH,37H,20H,7CH,60H,0EEH,0C0H
B80F 7C60EEC0
B813 547FEA80           DB      54H,7FH,0EAH,80H,0,0AH,0FEH,8
```

```
B817 000AFE08
B81B 0000AAA0          DB      0,0,0AAH,0A0H,0,0,0,0
B81F 00000000
B823 00000000 GMENS:   DB      0,0,0,0,0,1FH,0FCH,0        ・G面のデータ
B827 001FFC00
B82B 00FFFF80          DB      0,0FFH,0FFH,80H,3,0FH,0FH,0E0H
B82F 030F0FE0
B833 06F6F7F0          DB      6,0F6H,0F7H,0F0H,0DH,0F9H,0FBH,0F8H
B837 0DF9FBF8
B83B 0DC93BFC          DB      0DH,0C9H,3BH,0FCH,0EH,0F6H,0F7H,0FCH
B83F 0EF6F7FC
B843 0F0F0FFC          DB      0FH,0FH,0FH,0FCH,1FH,0FFH,0FFH,0FCH
B847 1FFFFFFC
B84B 3B9F3778          DB      3BH,09FH,37H,78H,7DH,0E0H,0EEH,0F0H
B84F 7DE0EEF0
B853 54FFEAC2          DB      54H,0FFH,0EAH,0C2H,0,1FH,0FFH,0CH
B857 001FFF0C
B85B 0001FFF0          DB      0,1,0FFH,0F0H,0,0,0,0
B85F 00000000
```

色盲検査表のような背景模様の中央に、本書のマスコット・キャラクタである『おばけのピーヨ』が現れました。プログラムは、例の『ANDを取ってORを取る』という重ね合わせの基本そのものです。キャラクタ・データは透明（くり抜き）＋B面＋R面＋G面の順で並んでいます。最初に透明データによってキャラクタ部分をくり抜いていますが、このときの透明データ構造は図2-2-1のようになっています。



黒い部分のビット＝1(背景が残る)
白い部分のビット＝0(くり抜かれる)

図2-2-1　透明データの構造(LIST 2.2用)

速度を追求するのであれば、キャラクタ・データのポインタにはインデックス・レジスタよりも裏レジスタ（HL）を用いたほうが速くなります。裏レジスタの用法に自信のない方は、ちょうどいい機会ですからやってみてください。ただし、小さなキャラクタ1ケではその速さを目で確認するのは無理ですが……。

では、次に同じことをALUを利用してやってみましょう。といっても、まったく同じではツマラナイので、せめてもの工夫で背景の模様を変えてみました。

CLEAR,&HB5FF:A=&HB700:CALL A

LIST 2.3

```
                             ORG     0B700H

B700 F3                      DI                       ・割込み禁止
B701 3EAA                    LD      A,10101010B      ┐
B703 D35C                    OUT     (5CH),A          │
B705 CD41B7                  CALL    BACKD            │
B708 AF                      XOR     A                │
B709 D35D                    OUT     (5DH),A          │・背景作成
B70B CD41B7                  CALL    BACKD            │
B70E 3D                      DEC     A                │
B70F D35E                    OUT     (5EH),A          │
B711 CD41B7                  CALL    BACKD            │
B714 D35F                    OUT     (5FH),A          ┘
B716 DB32                    IN      A,(32H)          ┐
B718 CBF7                    SET     6,A              │・拡張アクセスモード
B71A D332                    OUT     (32H),A          ┘
B71C 3E80                    LD      A,10000000B      ┐・G.VRAMをセレクト（ALUモード）
B71E D335                    OUT     (35H),A          ┘
B720 2100BF                  LD      HL,TMEN0         ┐
B723 AF                      XOR     A                │・キャラクタ部分をくり抜く
B724 CD55B7                  CALL    DICHR            ┘
B727 3E67                    LD      A,01100111B      ┐
B729 CD55B7                  CALL    DICHR            │
B72C 3E57                    LD      A,01010111B      │・おばけのピーヨ表示
B72E CD55B7                  CALL    DICHR            │
B731 3E37                    LD      A,00110111B      │
B733 CD55B7                  CALL    DICHR            ┘
B736 AF                      XOR     A                ┐・メインメモリをセレクト
B737 D335                    OUT     (35H),A          ┘
B739 DB32                    IN      A,(32H)          ┐
B73B CBB7                    RES     6,A              │・独立アクセスモード
B73D D332                    OUT     (32H),A          ┘
B73F FB                      EI                       ・割込み許可
B740 C9                      RET                      ・BASICへ戻る

B741 2100C0       BACKD:     LD      HL,0C000H        ┐
B744 0E32                    LD      C,50             │
B746 06A0         BACL0:     LD      B,80*2           │
B748 77           BACL1:     LD      (HL),A           │
B749 23                      INC     HL               │
B74A 2F                      CPL                      │
B74B 77                      LD      (HL),A           │
B74C 23                      INC     HL               │・G.VRAM全体をAレジスタで埋める（毎回データを
B74D 2F                      CPL                      │ 反転）
B74E 10F8                    DJNZ    BACL1            │
B750 2F                      CPL                      │
B751 0D                      DEC     C                │
B752 20F2                    JR      NZ,BACL0         │
B754 C9                      RET                      ┘

B755 D334         DICHR:     OUT     (34H),A            ┐
B757 11D0DD                  LD      DE,0DDD0H          │
B75A 01FF10                  LD      BC,16*100H+0FFH    │
B75D EDA0 *04     DICLP:     LDI     4                  │
B765 3E4C                    LD      A,80-4             │
B767 83                      ADD     A,E                │・ALUを指定のモードにしてキャラクタデータを
B768 3001                    JR      NC,ADE80           │ セット
B76A 14                      INC     D                  │
B76B 5F           ADE80:     LD      E,A                │
B76C 10EF                    DJNZ    DICLP              │
B76E C9                      RET                        ┘

                             ORG     0BF00H
                                                          ・くり抜き用透明データ（ALU用）
BF00 001FFC00     TMEN0:     DB      0,1FH,0FCH,0,0,0FFH,0FFH,80H  ↓
```

```
BF04 00FFFF80
BF08 03FFFFE0           DB      3,0FFH,0FFH,0E0H,7,0FFH,0FFH,0F0H
BF0C 07FFFFF0
BF10 0FFFFFF8           DB      0FH,0FFH,0FFH,0F8H,1FH,0FFH,0FFH,0FCH
BF14 1FFFFFFC
BF18 1FFFFFFE           DB      1FH,0FFH,0FFH,0FEH,1FH,0FFH,0FFH,0FEH
BF1C 1FFFFFFE
BF20 1FFFFFFE           DB      1FH,0FFH,0FFH,0FEH,3FH,0FFH,0FFH,0FEH
BF24 3FFFFFFE
BF28 7FFFFFFC           DB      7FH,0FFH,0FFH,0FCH,0FFH,0FFH,0FFH,0FAH
BF2C FFFFFFFA
BF30 FFFFFFFF           DB      0FFH,0FFH,0FFH,0FFH,7CH,0FFH,0FFH,0FEH
BF34 7CFFFFFE
BF38 001FFFFC           DB      0,1FH,0FFH,0FCH,0,1,0FFH,0F0H
BF3C 0001FFF0
BF40 00000000 BMENS:    DB      0,0,0,0,0,1FH,0FCH,0  ・B面のデータ
BF44 001FFC00
BF48 00FFFF80           DB      0,0FFH,0FFH,80H,3,0FFH,0FFH,0E0H
BF4C 03FFFFE0
BF50 070F0FF0           DB      7,0FH,0FH,0F0H,0EH,6,7,0F8H
BF54 0E0607F8
BF58 0E0607FC           DB      0EH,6,7,0FCH,0FH,0FH,0FH,0FCH
BF5C 0F0F0FFC
BF60 0FFFFFFC           DB      0FH,0FFH,0FFH,0FCH,1FH,0FFH,0FFH,0FCH
BF64 1FFFFFFC
BF68 2B9F3278           DB      2BH,09FH,32H,78H,55H,0E0H,0E4H,0F0H
BF6C 55E0E4F0
BF70 00FFE0E2           DB      0,0FFH,0E0H,0E2H,0,1FH,0FFH,8CH
BF74 001FFF8C
BF78 0001FFF0           DB      0,1,0FFH,0F0H,0,0,0,0
BF7C 00000000
BF80 00000000 RMENS:    DB      0,0,0,0,0,6,0D0H,0   ・R面のデータ
BF84 0006D000
BF88 003FFA00           DB      0,3FH,0FAH,0,1,0FH,0EH,80H
BF8C 010F0E80
BF90 02F6F740           DB      2,0F6H,0F7H,40H,5,0F9H,0FBH,0A0H
BF94 05F9FBA0
BF98 05C93BD0           DB      5,0C9H,3BH,0D0H,2,0F6H,0F7H,0A0H
BF9C 02F6F7A0
BFA0 070F0FD0           DB      7,0FH,0FH,0D0H,1BH,0FFH,0FFH,0A0H
BFA4 1BFFFFA0
BFA8 399F3720           DB      39H,09FH,37H,20H,7CH,60H,0EEH,0C0H
BFAC 7C60EEC0
BFB0 547FEA80           DB      54H,7FH,0EAH,80H,0,0AH,0FEH,8
BFB4 000AFE08
BFB8 0000AAA0           DB      0,0,0AAH,0A0H,0,0,0,0
BFBC 00000000
BFC0 00000000 GMENS:    DB      0,0,0,0,0,1FH,0FCH,0  ・G面のデータ
BFC4 001FFC00
BFC8 00FFFF80           DB      0,0FFH,0FFH,80H,3,0FH,0FH,0E0H
BFCC 030F0FE0
BFD0 06F6F7F0           DB      6,0F6H,0F7H,0F0H,0DH,0F9H,0FBH,0F8H
BFD4 0DF9FBF8
BFD8 0DC93BFC           DB      0DH,0C9H,3BH,0FCH,0EH,0F6H,0F7H,0FCH
BFDC 0EF6F7FC
BFE0 0F0F0FFC           DB      0FH,0FH,0FH,0FCH,1FH,0FFH,0FFH,0FCH
BFE4 1FFFFFFC
BFE8 3B9F3778           DB      3BH,09FH,37H,78H,7DH,0E0H,0EEH,0F0H
BFEC 7DE0EEF0
BFF0 54FFEAC2           DB      54H,0FFH,0EAH,0C2H,0,1FH,0FFH,0CH
BFF4 001FFF0C
BFF8 0001FFF0           DB      0,1,0FFH,0F0H,0,0,0,0
BFFC 00000000
```

すでに気がついているかもしれませんが、透明部分のキャラクタ・データが前回と違っています。ALUのビットリセット（ANDを取る）モードでは、データを反転して論理演算を行いますから、透明データも反転したものを用意しておかなければなりません。もちろん、ANDを取る前にCPL命令で反転することもできますが、それは実行速度を不用意に遅くするので非現実的です。結局、データ作成の時点からPC-8801mkⅡ以前／PC-8801SR以降という区別が必要になるわけです。

実際には、データ作成そのものはパターンエディタの役目ですから、どちらも工程は同じです。ただし、旧タイプのパターンエディタではALU用の透明データが作成できないものもあります。昔のパターンを利用する場合は『PTER88』でデータを作り直すか、最初にプログラムで反転してからキャラクタ・データとしたほうがいいでしょう。ゲーム中にいちいち反転するのは、速度的に余計な負担となるので避けるべきです。

なお、今回の『おばけのピーヨ』データは、この先いろいろなプログラムで利用しますが、今回以外は本書のプログラムリストからは省いてあります。

ここで、ダメージを受けたときなどによく使われるカラー反転表示を、ALUの残った機能（ビット反転）を使って行ってみます。ビット反転（XORを取る）といっても、元データのビットが1でなければビットセット（ORを取る）と結果は同じです。

```
元データ　:          00000000B          11111111B
デ ー タ　:      XOR 11100111B      XOR 11100111B
------------------------------------------------
新データ　:          11100111B          00011000B
```

したがって、重ね合わせたキャラクタが反転カラーになるためには、透明データでくり抜く部分を逆に埋めなければなりません。「くり抜く／埋める」を交互に行えば「ノーマル表示／反転カラー表示」を繰り返すことになり、ダメージなどの表現が簡単にできるわけです。**LIST 2.4**は、スペースキーを押すたびに表示が変わるよう、透明データ使用時のモード（ALUの設定）をAND／ORと切り換えています。この切り換えにもXORを利用しています（CKXORの次の行）。では、『おばけのピーヨ』データをロードして実行してみてください（BSAVEされたプログラムには、データも含まれています）。

```
CLEAR,&HB5FF:BLOAD"PIYO"
A=&HB700:CALL A
```

LIST 2.4

```
                          ORG     0B700H

BF00=            TMEN0:   EQU     0BF00H              ┐
BF40=            BMENS:   EQU     TMEN0+4*16          │・キャラクタデータアドレス
BF80=            RMENS:   EQU     BMENS+4*16          │
BFC0=            GMENS:   EQU     RMENS+4*16          ┘

B700 F3                   DI                          ・割込み禁止
B701 3EF0                 LD      A,11110000B         ┐
B703 D35C                 OUT     (5CH),A             │
B705 CD59B7               CALL    BACKD               │
B708 3E33                 LD      A,00110011B         │
B70A D35D                 OUT     (5DH),A             │
B70C CD59B7               CALL    BACKD               │・背景作成
B70F 3E55                 LD      A,01010101B         │
B711 D35E                 OUT     (5EH),A             │
B713 CD59B7               CALL    BACKD               │
B716 D35F                 OUT     (5FH),A             ┘
B718 DB32                 IN      A,(32H)             ┐
B71A CBF7                 SET     6,A                 │・拡張アクセスモード
B71C D332                 OUT     (32H),A             ┘
B71E 3E80                 LD      A,10000000B         ┐・G.VRAMをセレクト (ALUモード)
B720 D335                 OUT     (35H),A             ┘

B722 2100BF      DLOOP:   LD      HL,TMEN0            ┐
B725 3E07        CKXOR:   LD      A,00000111B         │
B727 EE07                 XOR     00000111B           │
B729 3226B7               LD      (CKXOR+1),A         │
B72C CD6CB7               CALL    DICHR               │
B72F 3E76                 LD      A,01110110B         │・キャラクタ表示（ノーマル／反転）を交互に行う
B731 CD6CB7               CALL    DICHR               │
B734 3E75                 LD      A,01110101B         │
B736 CD6CB7               CALL    DICHR               │
B739 3E73                 LD      A,01110011B         │
B73B CD6CB7               CALL    DICHR               ┘
B73E DB09        KSLP0:   IN      A,(9)               ┐
B740 07                   RLCA                        │
B741 07                   RLCA                        │・[SPACE] が押されていればループ
B742 30FA                 JR      NC,KSLP0            ┘
B744 DB09        KSLP1:   IN      A,(9)               ┐
B746 07                   RLCA                        │・[ESC] が押されていればTOBSCへ
B747 3005                 JR      NC,TOBSC            ┘
B749 07                   RLCA                        ┐・[SPACE] が押されていればDLOOPへ
B74A 30D6                 JR      NC,DLOOP            ┘
B74C 18F6                 JR      KSLP1               ・KSLP1へ
B74E AF          TOBSC:   XOR     A                   ┐・メインメモリをセレクト
B74F D335                 OUT     (35H),A             ┘
B751 DB32                 IN      A,(32H)             ┐
B753 CBB7                 RES     6,A                 │・独立アクセスモード
B755 D332                 OUT     (32H),A             ┘
B757 FB                   EI                          ・割込み許可
B758 C9                   RET                         ・BASICへ戻る

B759 2100C0      BACKD:   LD      HL,0C000H           ┐
B75C 0E64                 LD      C,100               │
B75E 06A0        BACL0:   LD      B,80*2              │
B760 77          BACL1:   LD      (HL),A              │
B761 0F                   RRCA                        │
B762 0F                   RRCA                        │・背景の表示
B763 23                   INC     HL                  │
B764 10FA                 DJNZ    BACL1               │
B766 0F                   RRCA                        │
B767 0F                   RRCA                        │
```

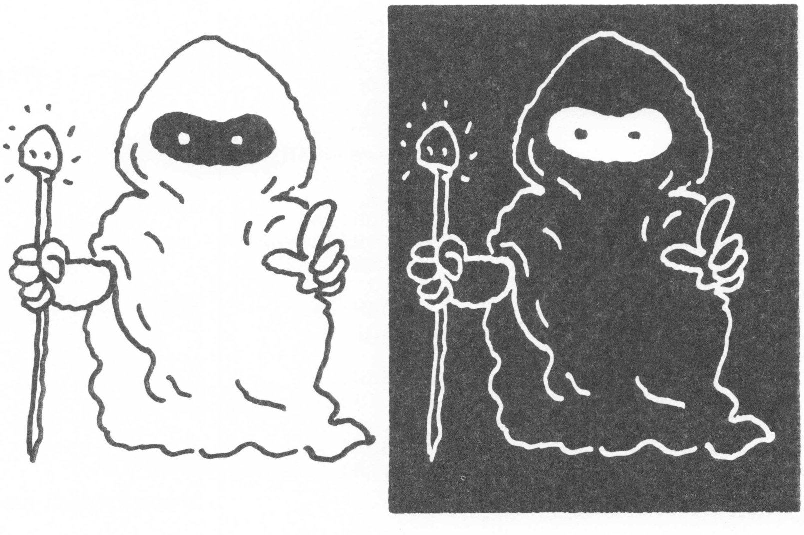
```
B768 0D                  DEC     C
B769 20F3                JR      NZ,BACL0
B76B C9                  RET

B76C D334       DICHR:   OUT     (34H),A
B76E 11D0DD              LD      DE,0DDD0H
B771 01FF10              LD      BC,16*100H+0FFH
B774 EDA0 *04   DICLP:   LDI     4
B77C 3E4C                LD      A,80-4
B77E 83                  ADD     A,E
B77F 3001                JR      NC,ADE80
B781 14                  INC     D
B782 5F         ADE80:   LD      E,A
B783 10EF                DJNZ    DICLP
B785 C9                  RET
```

・ALUを指定のモードにしてキャラクタデータをセット

このプログラムでは、反転カラー時にくり抜き部を3面とも埋めていますが、これを面によって変えるとまた違った反転カラーキャラクタを作ることができます。変更の方法は簡単で、(CKXOR＋1) の値をカラー番号と同じように指定（例えば、5ならB／G面のみ反転）すればいいのです。ただし、そのときにはXORを取る値（次の行）も同じにしなければなりません。

このあたりの変化は、文章で一つひとつ説明するとキリがありませんから、いろいろと適当に数値を変化させながら実験してください。それによって、ALUのモードの違いや利用法が自然と理解できるようになるでしょう。



2-3 独立アクセスモード vs. ALU

高級品は普及品に用意されている機能をすべて持っているかというと、意外とそうでもありません。普及品はあくまでも価格が普及版という意味で、実際には上位のものに負けないよう頑張っているのです。腕時計など、高級品より普及品のほうが機能が多いこともあるくらいです。こうなると、時と場合によって使い分けるようなセンスも要求されているといえるでしょう。

ALUという多機能なモードの存在を知ると、古くからの独立アクセスモードなど不要に思えてきます。しかし、実際には独立アクセスモードならではの特徴もあり、ALUより便利なときもあるのです。温故知新というには余りに大袈裟ですが、独立アクセスモードの短所（論理演算が自動的にできない）を再認識することは、ALUの長所（論理演算が自動的にできる）を熟知することにつながります。もしかすると、『短所は長所であり長所は短所である』ことを発見できるかもしれません。

そんな格言を実践してみたのが、次のプログラム（LIST 2.5）です。では、早速故きを温ねて新しきを知ってみましょう。

```
CLEAR,&HB5FF:A=&HB700:CALL A
```

LIST 2.5

```
                          ORG     0B700H

B700 F3                   DI                      ・割込み禁止
B701 11A000               LD      DE,80*2        ┐
B704 217AB7               LD      HL,SUJI0       │
B707 DD21D0DD   TLOOP:    LD      IX,0DDD0H      │
B70B D35D                 OUT     (5DH),A        ├・独立アクセスモードで数字を表示
B70D CD51B7               CALL    WSDIP          │
B710 D35F                 OUT     (5FH),A        ┘
B712 DD21D4DD             LD      IX,0DDD4H      ┐
B716 DB32                 IN      A,(32H)        │
B718 CBF7                 SET     6,A            │
B71A D332                 OUT     (32H),A        │
B71C 3E80                 LD      A,10000000B    │
B71E D335                 OUT     (35H),A        │
B720 3E02                 LD      A,2            │
B722 D334                 OUT     (34H),A        ├・拡張アクセスモードで数字を表示
B724 CD51B7               CALL    WSDIP          │
B727 AF                   XOR     A              │
B728 D335                 OUT     (35H),A        │
B72A DB32                 IN      A,(32H)        │
B72C CBB7                 RES     6,A            │
B72E D332                 OUT     (32H),A        ┘
B730 DB09       KSLP0:    IN      A,(9)          ┐
B732 07                   RLCA                   │
B733 07                   RLCA                   ├・[SPACE] が押されていればループ
B734 30FA                 JR      NC,KSLP0       ┘
B736 DB09       KSLP1:    IN      A,(9)          ┐
B738 07                   RLCA                   ├・[ESC] が押されていればTLOOPへ
B739 3014                 JR      NC,TOBSC       ┘
B73B 07                   RLCA                   ┐
B73C 38F8                 JR      C,KSLP1        ┘・[SPACE] が押されていなければKSLP1へ
```

```
B73E 217AB7            LD      HL,SUJI0
B741 3E00     CKS01:   LD      A,0
B743 EE01              XOR     1
B745 3242B7            LD      (CKS01+1),A     ;・HLに0／1のデータアドレスを交互に指定して
B748 28BD              JR      Z,TLOOP         ;  TLOOPへ
B74A 2182B7            LD      HL,SUJI1
B74D 18B8              JR      TLOOP
B74F FB       TOBSC:   EI                      ;・割込み許可
B750 C9                RET                     ;・BASICへ戻る

B751 E5       WSDIP:   PUSH    HL
B752 0E08              LD      C,8
B754 0604     WSDLP:   LD      B,4
B756 CB0E     WLOLP:   RRC     (HL)
B758 1F                RRA
B759 CB2F              SRA     A
B75B 10F9              DJNZ    WLOLP
B75D DD7701            LD      (IX+1),A
B760 DD7751            LD      (IX+81),A
B763 0604              LD      B,4
B765 CB0E     WHILP:   RRC     (HL)            ;・データを縦横2倍に拡大して表示
B767 1F                RRA
B768 CB2F              SRA     A
B76A 10F9              DJNZ    WHILP
B76C DD7700            LD      (IX+0),A
B76F DD7750            LD      (IX+80),A
B772 23                INC     HL
B773 DD19              ADD     IX,DE
B775 0D                DEC     C
B776 20DC              JR      NZ,WSDLP
B778 E1                POP     HL
B779 C9                RET

B77A 7E       SUJI0:   DB      01111110B
B77B C3                DB      11000011B
B77C C3                DB      11000011B
B77D C3                DB      11000011B       ;・数字0のデータ
B77E C3                DB      11000011B
B77F C3                DB      11000011B
B780 C3                DB      11000011B
B781 7E                DB      01111110B

B782 18       SUJI1:   DB      00011000B
B783 38                DB      00111000B
B784 18                DB      00011000B
B785 18                DB      00011000B
B786 18                DB      00011000B       ;・数字1のデータ
B787 18                DB      00011000B
B788 18                DB      00011000B
B789 3C                DB      00111100B
```

実行すると、画面中央に‘0’が2つ並んで赤く表示されます。左側が独立アクセスモードでの表示で、右側がALUのビットセット（ORを取る）による表示です。ここでスペースキーを押すたびに‘1’と‘0’を交互に繰り返すように表示させたいのですが、結果はどうでしょうか……。

左側の数字は押すたびにキチンと交互に表示されますが、右側は2つの数字が重なって表示されてしまい変化になりません。つまり、ALUでは常に論理演算がなされるため、データをそのままの姿（0のビットは0で、1のビットは1で）ではグラフィックVRAMにストアできないのです。図2-1-3に「ノーオペレーション」というモードがありますが、これはデータに対して何もしないのではなく、グラフィックVRAMに対して何もしないという意味ですから、この場合は役に立ちません。どうしてもALUを利用したければ、いったんビットリセット（ANDを取る）

モードでFFHをストアし、表示されている数字をクリアしなければならないわけです。

結局、ALUの機能というのは重ね合わせ処理には威力を発揮しますが、そうでない場合には欠点になる危険性もあるということです。例えば、重ね合わせ処理をしない旧タイプのゲーム（背景＝黒）では、移動の際にキャラクタの残骸部分を消去しますが、ALUによる表示では図2-3-1のように困った現象が起きてしまいます。

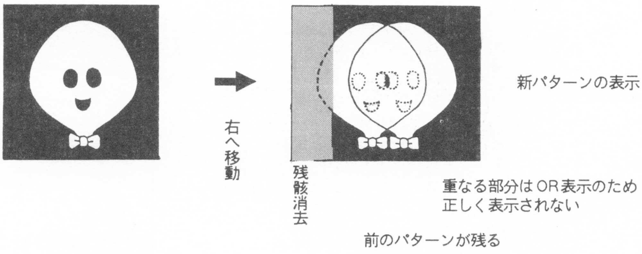


図2-3-1　キャラクタの移動をALU(ビットセット)で行うと‥‥

これを避けるためには、キャラクタを毎回全消去すればいいのですが、今度はキャラクタがチカチカして非常に見にくくなります。したがって、このような場合は素直に独立アクセスモードを利用するほうがいいわけです。プログラミングに無駄な時間を費やさないためには、こういった使い分けが即断できるよう機能の特徴を正確に把握しておくことが大切なのです。

さて、このプログラムにはもう1つ小さなテクニックが隠されています。それは、パターンの拡大表示（ここでは縦横倍角）です。縦を倍にするのは、同じデータを一段下のグラフィックVRAMに入れるだけですが、横を倍にするのはデータ作成が少し面倒です。それは、図2-3-2のようにデータを2つに分けて拡大しなければならないからです。

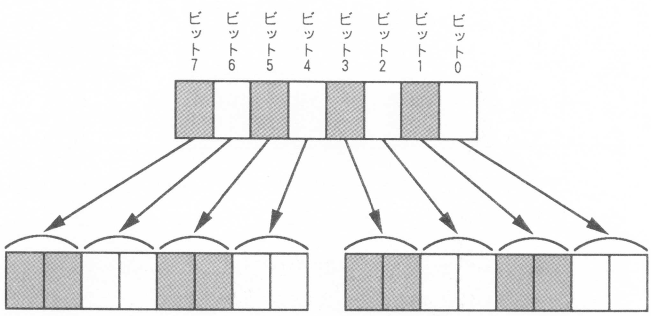


図2-3-2　データの拡大(横2倍)

ここでは、キャリーフラグとSRA命令というめったに使わない命令を活用し、横2倍を実現しています。下位4ビットから先にデータを作っているのは、このSRA命令の特徴によるものですが、ビットの動きさえ理解できれば横3倍でも横4倍でも自由に拡大できるはずです。いろいろと応用のきくテクニックですから、参考ついでにトライしてみてください。

2-4 3面同時リード/ライト

簡単なことでも、同じことを3度繰り返すのは面倒なものです。もしも言葉を正確にするという妙な理由で、『今日からすべての文字は3度繰り返すこと』などという法律ができとしたら……。想像しただけでも大変です。

「こここんんんにににちちちわわわ」

「やややあああ、、、どどどうううももも」

こんなバカなことが現実に起こるとは思えませんが、ここコンピュータ88国のグラフィックVRAM地方では、それに近いことをしなければなりません。画面を作成するときは、各プレーンにバラバラの値を入れるわけですから、同じような作業が続いてもある程度は仕方のないことと納得できます。しかし、いったん画面に絵として描かれたなら、プレーンという感覚はなくなるのが普通です。例えば、グラフィック画面をロールアップしたりダウンしたりする場合、プレーンごとに同じ作業（データの転送）を3度繰り返すというのは当然とはいえ面倒なことです。

こんなとき便利なのが、拡張グラフィックスの『直前のリードサイクルでリードされたデータを3面同時にライトする』というモードです（図2-1-2)。これは、3画面分のデータを一括して転送する機能ですが、アイデア次第で利用法は大きく広がります。ただ残念なことに、リードしたデータに手を加えること（ANDを取って部分消去する……等）はできません。あくまでも1バイト単位でリードしたものを、そのままの状態でライトするだけです。

参考例として、この3面同時転送機能を使ってグラフィック画面を縦横自在に移動（横は1バイト単位）させるプログラム（LIST 2.6）を紹介します。これは、テンキー（2 4 6 8）の操作で画面を上下左右にスクロール・ループさせるものです。しかし、左右だけ8ドット（1バイト）単位では不満が残るかもしれませんね。そこで、SHIFTキーを押すと左右も1ドット単位で動くようにしてあります。ただし、これは独立アクセスモードでしか実現できませんので、そのつもりでプログラムを見てください。なお、実行前にグラフィック画面に何か描いておかないと、闇夜のカラスになってしまいます。



```
LINE(0,0)-(639,199),7:LINE(0,199)-(639,0),5
CLEAR,&HB5FF:A=&HB700:CALL A
```

LIST 2.6

```
                              ORG     0B700H

B700 F3                       DI                        ・割込み禁止
B701 2101B7         TLOOP:    LD      HL,TLOOP          ・リターンアドレスをTLOOPにセット
B704 E5                       PUSH    HL
B705 DB00                     IN      A,(0)
B707 CB57                     BIT     2,A               ・[2] が押されていればMOVE2へ
B709 CA2FB7                   JP      Z,MOVE2
B70C CB67                     BIT     4,A               ・[4] が押されていればMOVE4へ
B70E CA4FB7                   JP      Z,MOVE4
B711 CB77                     BIT     6,A               ・[6] が押されていればMOVE6へ
B713 CAA4B7                   JP      Z,MOVE6
B716 DB01                     IN      A,(1)
B718 0F                       RRCA                      ・[8] が押されていればMOVE8へ
B719 D2F9B7                   JP      NC,MOVE8
B71C E1                       POP     HL                ・ダミー
B71D DB09                     IN      A,(9)
B71F 07                       RLCA                      ・[ESC] が押されていなければTLOOPへ
B720 38DF                     JR      C,TLOOP
B722 FB                       EI                        ・割込み許可
B723 C9                       RET                       ・BASICへ戻る

B724 DB32           KGMOD:    IN      A,(32H)           ・拡張アクセスモード
B726 CBF7                     SET     6,A
B728 D332                     OUT     (32H),A
B72A 3E90                     LD      A,10010000B       ・三面同時リードライトモードでG.VRAMをセレクト
B72C D335                     OUT     (35H),A             してリターン
B72E C9                       RET

B72F CD24B7         MOVE2:    CALL    KGMOD
B732 217FFE                   LD      HL,0FE7FH
B735 11CFFE                   LD      DE,0FE7FH+80
B738 01803E                   LD      BC,80*200
B73B EDB8                     LDDR                      ・グラフィック画面を下方向へループスクロール
B73D 21CFFE                   LD      HL,0FE7FH+80
B740 015000                   LD      BC,80
B743 EDB8                     LDDR
B745 AF             NMMOD:    XOR     A
B746 D335                     OUT     (35H),A
B748 DB32                     IN      A,(32H)
B74A CBB7                     RES     6,A               ・メインメモリ／独立アクセスモードをセレクトして
B74C D332                     OUT     (32H),A             リターン
B74E C9                       RET

B74F DB08           MOVE4:    IN      A,(8)
B751 07                       RLCA                      ・[SHIFT] が押されていればMDOT4へ
B752 07                       RLCA
B753 3021                     JR      NC,MDOT4
B755 CD24B7                   CALL    KGMOD
B758 2101C0                   LD      HL,0C001H
B75B 1100C0                   LD      DE,0C000H
B75E 06C8                     LD      B,200
B760 C5             MV4LP:    PUSH    BC
B761 1A                       LD      A,(DE)
B762 3280FE                   LD      (0FE80H),A
B765 014F00                   LD      BC,79             ・グラフィック画面を左方向へループスクロール
B768 EDB0                     LDIR
B76A 3A80FE                   LD      A,(0FE80H)
B76D 12                       LD      (DE),A
B76E 23                       INC     HL
B76F 13                       INC     DE
B770 C1                       POP     BC
B771 10ED                     DJNZ    MV4LP
B773 C345B7                   JP      NMMOD
```

```
B776 D35C      MDOT4:  OUT    (5CH),A
B778 CD88B7            CALL   MD4SB
B77B D35D              OUT    (5DH),A
B77D CD88B7            CALL   MD4SB
B780 D35E              OUT    (5EH),A
B782 CD88B7            CALL   MD4SB
B785 D35F              OUT    (5FH),A
B787 C9                RET
B788 217FFE    MD4SB:  LD     HL,0FE7FH
B78B 1130FE            LD     DE,0FE7FH-79
B78E 06C8              LD     B,200
B790 C5        MDS4L:  PUSH   BC
B791 1A                LD     A,(DE)        ・ドット単位でグラフィック画面を左方向へループス
B792 07                RLCA                  クロール
B793 0650              LD     B,80
B795 CB16      M4RLP:  RL     (HL)
B797 2B                DEC    HL
B798 10FB              DJNZ   M4RLP
B79A EB                EX     DE,HL
B79B 01B0FF            LD     BC,-80
B79E 09                ADD    HL,BC
B79F EB                EX     DE,HL
B7A0 C1                POP    BC
B7A1 10ED              DJNZ   MDS4L
B7A3 C9                RET

B7A4 DB08      MOVE6:  IN     A,(8)
B7A6 07                RLCA                 ・[SHIFT] が押されていればMDOT6へ
B7A7 07                RLCA
B7A8 3021              JR     NC,MDOT6
B7AA CD24B7            CALL   KGMOD
B7AD 217EFE            LD     HL,0FE7EH
B7B0 117FFE            LD     DE,0FE7FH
B7B3 06C8              LD     B,200
B7B5 C5        MV6LP:  PUSH   BC
B7B6 1A                LD     A,(DE)
B7B7 3280FE            LD     (0FE80H),A
B7BA 014F00            LD     BC,79
B7BD EDB8              LDDR                 ・グラフィック画面を右方向へループスクロール
B7BF 3A80FE            LD     A,(0FE80H)
B7C2 12                LD     (DE),A
B7C3 2B                DEC    HL
B7C4 1B                DEC    DE
B7C5 C1                POP    BC
B7C6 10ED              DJNZ   MV6LP
B7C8 C345B7            JP     NMMOD

B7CB D35C      MDOT6:  OUT    (5CH),A
B7CD CDDDB7            CALL   MD6SB
B7D0 D35D              OUT    (5DH),A
B7D2 CDDDB7            CALL   MD6SB
B7D5 D35E              OUT    (5EH),A
B7D7 CDDDB7            CALL   MD6SB
B7DA D35F              OUT    (5FH),A
B7DC C9                RET
B7DD 2100C0    MD6SB:  LD     HL,0C000H
B7E0 114FC0            LD     DE,0C000H+79
B7E3 06C8              LD     B,200
B7E5 C5        MDS6L:  PUSH   BC
B7E6 1A                LD     A,(DE)        ・ドット単位でグラフィック画面を右方向へループス
B7E7 0F                RRCA                  クロール
B7E8 0650              LD     B,80
B7EA CB1E      M6RLP:  RR     (HL)
B7EC 23                INC    HL
B7ED 10FB              DJNZ   M6RLP
B7EF EB                EX     DE,HL
B7F0 015000            LD     BC,80
B7F3 09                ADD    HL,BC
B7F4 EB                EX     DE,HL
B7F5 C1                POP    BC
```

```
B7F6 10ED              DJNZ    MDS6L
B7F8 C9                RET

B7F9 CD24B7   MOVE8:   CALL    KGMOD
B7FC 2100C0            LD      HL,0C000H
B7FF 1180FE            LD      DE,0FE80H
B802 015000            LD      BC,80
B805 EDB0              LDIR                    ・グラフィック画面を上方向へループスクロール
B807 1100C0            LD      DE,0C000H
B80A 01803E            LD      BC,80*200
B80D EDB0              LDIR
B80F C345B7            JP      NMMOD
```

このプログラムは速度の追求はしていませんが、LDIR／LDDR命令の代わりにLDI／LDD命令を連続したり、ループをやめて命令を80組連続させると（ドット移動時）、より高速にプログラムを実行することができます。状況に合わせ、そういった速度アップのテクニックも組み合わせるといいでしょう。

先ほど3面同時転送モードではデータを操作できないと書きましたが、これはリードされた3画面分のデータが実際にはそのレジスタに入っていないからです。通常のレジスタには1バイトしか記憶できませんから、3画面分のデータを記憶できないのは当然のことですね。つまり、このモードでグラフィックVRAMをリードした場合、3画面分のデータはまったく別のところに記憶され、次にグラフィックVRAMへライトしたときにはそれが3画面に対して出力されるということです。

したがって、レジスタはグラフィックVRAMをアクセス（リード／ライト）するためのダミーです。例えば、プログラムの最後（MOVE8）で一番上のラインをFE80Hからに退避している部分がありますが、これは次のように変更してもちゃんと動作します。

```
LD    HL,0C000H        →              LD    HL,0C000H
LD    DE,0FE80H                       LD    DE,0FE80H
LD    BC,80                           LD    A,80
LDIR                         M8LOP:   LD    B,(HL)
                                      LD    (DE),A
                                      INC   HL
                                      INC   DE
                                      DEC   A
                                      JR    NZ,M8LOP
```

なんと、Bレジスタでリードして A レジスタでライトしています。しかも、Aレジスタはカウンタとして使用中ですから、常識的に考えればこのプログラムは意味のないものです。しかし、これが事実として成立するということは、逆に考えればレジスタの値を操作しても3画面分のデータを操作したことにならないということになります。

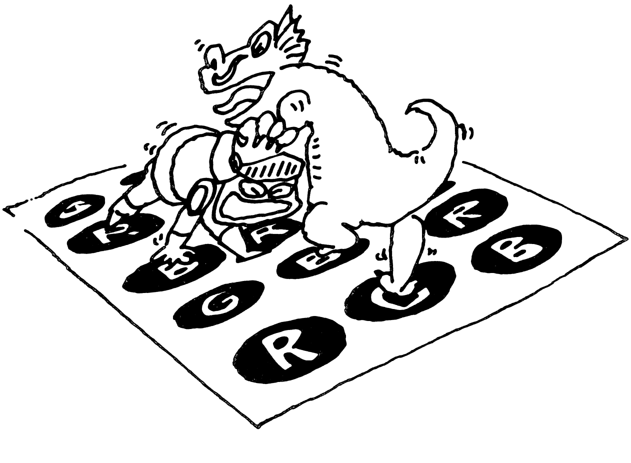
では、このときBレジスタも別の用途として使用できるのでしょうか。リード

したものがレジスタに入らないならば、Bレジスタの値も変わらないと考えられます。……が、結論から言えばこれはできません。Bレジスタには、3画面分のデータではなくまったく別のデータが入るからです。それは、この第2章の最初で触れたように、『拡張アクセスモードでグラフィックVRAMをリードすると、レジスタには指定カラーのビットを1にした値が入る』という機能のためです。本プログラムでは、ポート35Hへ10010000Bという値を出力していますから、リードするとカラー番号＝0のビットを1にした値が返ってきています。

なお、各移動ルーチン（MOVE2～MOVE8）へはジャンプして行っていますが、そこからメインルーチンへはRET命令で戻っています。これはメインルーチンの先頭で、戻りアドレス（TLOOP）をPUSHすることで実現しているスタック操作テクニックの一種です。簡単なテクニックですが、分岐先が多い場合には非常に有益です。単にメモリの節約になるだけでなく、各プログラムがサブルーチンとして他でも使用することができるようになるからです。グラフィック操作の陰に隠れていますが、こんなミニテクがあったことも忘れないでください。

グラフィック画面の転送というテーマのついでに、『直前でリードされたR（B）面のデータをB（R）面にライトする』というモードについて触れておきましょう。どういうわけかG面だけはカヤの外に置かれ、これだけでは何のために使用するのかわからないかもしれません。実は、これはゲームにはほとんど関係のない機能で、ワープロなどのためにあるモードです。図1-1-3を振り返れば、「アァ、そうか!!」と納得できるのではないでしょうか。画面の上半分から下半分への転送（またはその逆）などの際に、このモードで実行すればバンク切り換えが不要になります。とりあえず、存在と目的がわかればこのモードは卒業していいでしょう。

ちなみに、ここでリードしたデータは1バイト（1画面分）ですが、データはレジスタには入らず3面同時転送とまったく同じ状態になります。レジスタには、ポート35Hへ出力したカラー番号のビットを立てた値が入ります。つまり、価値としては400ラインモードのためにあるのですが、機能自体は特に400ライン専用というわけではなく、カラーモードでも同じ働きをするということです。興味があれば、カラーモードでの利用法を考えてみるのもいいかもしれません。



2-5 グラフィックVRAMの余り

日本人は余暇の利用が下手といわれていますが、中には遊ぶために遊び方を勉強するという人もいるそうです。また、それがイヤなために「無理に遊ぶくらいなら仕事をするほうがマシ!!」と言う人もいます。一番いいのは、遊びと仕事が同じになることですが、せっかくの遊びも仕事となると楽しんでばかりはいられなくなるようです。コンピュータなどその典型かもしれません……。

余暇の利用というわけではありませんが、前節のプログラム（LIST 2.6）ではグラフィックVRAMのFE80H以降を、グラフィック・データの一時退避用として大いに活用していました。このグラフィックVRAMのFE80H以降というのは、雰囲気的には単なる余りのような存在ですが、実は大変利用価値の高いメモリエリアなのです。というのは、同じアドレスに3画面分のデータを入れることができる上、その内容が画面に表示されないという特徴があるからです。

例えば、重ね合わせ処理をしたキャラクタ・パターンを動かす場合、実際には「背景でキャラクタを消去し、その上に新たなパターンを重ね合わせて表示する」という作業が必要です。

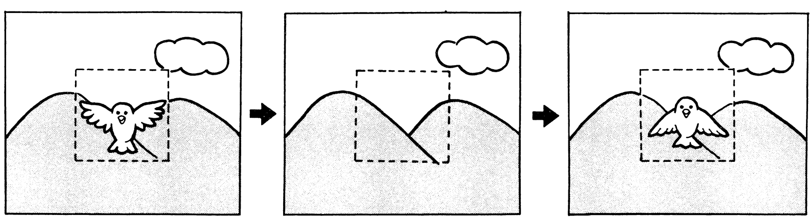


図2-5-1　重ね合わせ処理とアニメーションの原理

これをまともに実行すれば、背景でキャラクタを消去した時点で画面はキャラクタ不在の状態になります。もちろん、その直後には新たなキャラクタ・パターンが描かれるわけですが、背景データの構造／格納法、キャラクタのサイズ……等、条件によってはキャラクタ不在という事実がチラツキとなって現れてくることがあります。これを解決するためには、キャラクタの消去（背景の表示）と新キャラクタの表示を同時に行わなければなりません。つまり、『背景表示＋くり抜き（AND）＋キャラクタ（OR)』というデータ処理を、グラフィックVRAMにデータを入れる前に行うということです。

これはけっして不可能なことではありませんが、1つのキャラクタに対し背景パターンが複数で構成されている場合（ゲームではこのようなケースも多い)、レジスタが不足してプログラムは難解複雑になります。そこで、余っているグラフィックVRAMのFE80H以降を利用しようというのです。手順としては、図2-5-2のようなものになります。

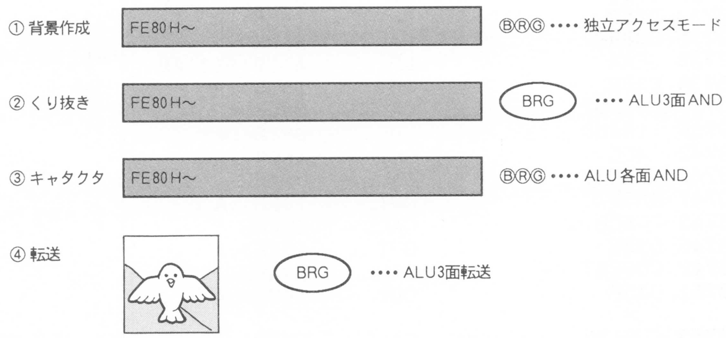


図2-5-2　グラフィックVRAMのFE80H以降の利用手順

FE80H以降への表示（実際には見えない）は、アドレスが連続しているので非常に簡単です（単なるデータの連続転送でよい)。次に、それをALUの3面同時転送機能を使って表示アドレスへ転送すれば、消去と表示の同時処理が実現できることになります。

では、これら3種類の「重ね合わせアニメ処理」を『おばけのピーヨ』を使って実験してみましょう。データをロードしてからプログラム（LIST 2.7）を実行してください（BSAVEされたプログラムにはデータも含まれています)。スペースキーを押している間、3匹のピーヨがイヤイヤするように動きます（表示位置は固定)。

```
CLEAR,&HB5FF:BLOAD"PIYO"
A=&HB700:CALL A
```

LIST 2.7

```
                        ORG     0B700H

BD00=           BKDAT:  EQU     0BD00H          ・背景データアドレス
BE00=           RVDAT:  EQU     0BE00H          ・左右反転データアドレス
BF00=           TMEN0:  EQU     0BF00H          ┐
BF40=           BMENS:  EQU     TMEN0+4*16      │・キャラクタデータアドレス
BF80=           RMENS:  EQU     BMENS+4*16      │
BFC0=           GMENS:  EQU     RMENS+4*16      ┘

B700 F3                 DI                      ・割込み禁止
B701 2100BF             LD      HL,TMEN0        ┐
B704 110400             LD      DE,4            │
B707 DD2100BE           LD      IX,RVDAT        │
B70B 0E40               LD      C,16*4          │
B70D CD82B7     DREVL:  CALL    REVRS           │
B710 DD7703             LD      (IX+3),A        │
B713 CD82B7             CALL    REVRS           │
B716 DD7702             LD      (IX+2),A        │・ピーヨの左右反転データ作成
B719 CD82B7             CALL    REVRS           │
B71C DD7701             LD      (IX+1),A        │
B71F CD82B7             CALL    REVRS           │
B722 DD7700             LD      (IX+0),A        │
B725 DD19               ADD     IX,DE           │
B727 0D                 DEC     C               │
B728 20E3               JR      NZ,DREVL        ┘
B72A 3EF0               LD      A,11110000B     ┐
B72C D35C               OUT     (5CH),A         │・背景を表示
B72E CD8BB7             CALL    BACKD           ┘
```

```
B731 3E55                LD     A,01010101B
B733 D35D                OUT    (5DH),A
B735 CD8BB7              CALL   BACKD
B738 3EAA                LD     A,10101010B       ・背景を表示・背景データを作成する・ピーヨ表示
B73A D35E                OUT    (5EH),A            (方法－2)
B73C CD8BB7              CALL   BACKD
B73F 1100BD              LD     DE,BKDAT
B742 D35C                OUT    (5CH),A
B744 CD9CB7              CALL   BMAKE
B747 D35D                OUT    (5DH),A
B749 CD9CB7              CALL   BMAKE             ・ピーヨ表示（方法－3）
B74C D35E                OUT    (5EH),A
B74E CD9CB7              CALL   BMAKE
B751 D35F                OUT    (5FH),A

B753 CDB1B7     DLOOP:   CALL   PUTP1             ・ピーヨ表示（方法－1）
B756 CD09B8              CALL   PUTP2             ・ピーヨ表示（方法－2）
B759 CD59B8              CALL   PUTP3             ・ピーヨ表示（方法－3）
B75C 3A81B7              LD     A,(PATAD+1)
B75F EE01                XOR    1                 ・パターン番号（1／0）を変更する
B761 3281B7              LD     (PATAD+1),A
B764 060A                LD     B,10
B766 DB40       VWAIT:   IN     A,(40H)
B768 E620                AND    00100000B
B76A 20FA                JR     NZ,VWAIT
B76C DB40       VWAT2:   IN     A,(40H)           ・ウェイト
B76E E620                AND    00100000B
B770 28FA                JR     Z,VWAT2
B772 10F2                DJNZ   VWAIT
B774 DB09       KSLP1:   IN     A,(9)
B776 07                  RLCA                     ・[ESC] が押されていればTOBSCへ
B777 3005                JR     NC,TOBSC
B779 07                  RLCA
B77A 30D7                JR     NC,DLOOP          ・[SPACE] が押されていればDLOOPへ
B77C 18F6                JR     KSLP1             ・KSLP1へ
B77E FB         TOBSC:   EI                       ・割込み許可
B77F C9                  RET                      ・BASICへ戻る

B780 00BF       PATAD:   DW     TMEN0             ・パターンアドレス

B782 0608       REVRS:   LD     B,8
B784 CB0E       REVL1:   RRC    (HL)
B786 17                  RLA
B787 10FB                DJNZ   REVL1             ・左右反転したデータを求める
B789 23                  INC    HL
B78A C9                  RET

B78B 2100C0     BACKD:   LD     HL,0C000H
B78E 0EC8                LD     C,200
B790 0650       BACL0:   LD     B,80
B792 77         BACL1:   LD     (HL),A
B793 0F                  RRCA
B794 0F                  RRCA
B795 23                  INC    HL                ・プレーン別に背景を表示
B796 10FA                DJNZ   BACL1
B798 0D                  DEC    C
B799 20F5                JR     NZ,BACL0
B79B C9                  RET

B79C 2100C0     BMAKE:   LD     HL,0C000H
B79F 3E10                LD     A,16
B7A1 EDA0 *04   BMAKL:   LDI    4
B7A9 014C00              LD     BC,80-4
B7AC 09                  ADD    HL,BC             ・プレーン別に背景データを作成
B7AD 3D                  DEC    A
B7AE 20F1                JR     NZ,BMAKL
B7B0 C9                  RET
```

```
B7B1 2100BD   PUTP1:  LD    HL,BKDAT          ┐
B7B4 D35C             OUT   (5CH),A           │
B7B6 CDF1B7           CALL  PBACK             │
B7B9 D35D             OUT   (5DH),A           │・背景を指定位置に表示
B7BB CDF1B7           CALL  PBACK             │
B7BE D35E             OUT   (5EH),A           │
B7C0 CDF1B7           CALL  PBACK             │
B7C3 D35F             OUT   (5FH),A           ┘
B7C5 DB32             IN    A,(32H)           ┐
B7C7 CBF7             SET   6,A               │
B7C9 D332             OUT   (32H),A           │
B7CB 3E80             LD    A,10000000B       │
B7CD D335             OUT   (35H),A           │
B7CF 2A80B7           LD    HL,(PATAD)        │
B7D2 AF               XOR   A                 │
B7D3 CDEFB7           CALL  PUT1D             │
B7D6 3E67             LD    A,01100111B       │
B7D8 CDEFB7           CALL  PUT1D             │
B7DB 3E57             LD    A,01010111B       │・ALUを使用してピーヨを指定位置に表示
B7DD CDEFB7           CALL  PUT1D             │
B7E0 3E37             LD    A,00110111B       │
B7E2 CDEFB7           CALL  PUT1D             │
B7E5 AF               XOR   A                 │
B7E6 D335             OUT   (35H),A           │
B7E8 DB32             IN    A,(32H)           │
B7EA CBB7             RES   6,A               │
B7EC D332             OUT   (32H),A           │
B7EE C9               RET                     ┘

B7EF D334     PUT1D:  OUT   (34H),A            ・ALUのモード設定
B7F1 117CDD   PBACK:  LD    DE,0DD7CH         ┐
B7F4 01FF10   PUTCC:  LD    BC,16*100H+0FFH   │
B7F7 EDA0 *04 PTCLP:  LDI   4                 │
B7FF 3E4C             LD    A,80-4            │
B801 83               ADD   A,E               │・指定位置にピーヨのデータをセット
B802 3001             JR    NC,ADE80          │
B804 14               INC   D                 │
B805 5F       ADE80:  LD    E,A               │
B806 10EF             DJNZ  PTCLP             │
B808 C9               RET                     ┘

B809 CD2CB8   PUTP2:  CALL  BKCPL              ・透明データを反転（ノーマル用となる）
B80C 1100BD           LD    DE,BKDAT          ┐
B80F D9               EXX                     │
B810 2A80B7           LD    HL,(PATAD)        │
B813 114000           LD    DE,4*16           │
B816 19               ADD   HL,DE             │
B817 014C00           LD    BC,80-4           │
B81A D9               EXX                     │
B81B D35C             OUT   (5CH),A           │・背景で消去しながら新たなピーヨを表示
B81D CD38B8           CALL  P2ALL             │
B820 D35D             OUT   (5DH),A           │
B822 CD38B8           CALL  P2ALL             │
B825 D35E             OUT   (5EH),A           │
B827 CD38B8           CALL  P2ALL             │
B82A D35F             OUT   (5FH),A           ┘
B82C 2A80B7   BKCPL:  LD    HL,(PATAD)        ┐
B82F 0640             LD    B,4*16            │
B831 7E       BKCLP:  LD    A,(HL)            │
B832 2F               CPL                     │
B833 77               LD    (HL),A            │・透明データを反転（ALU用となる）
B834 23               INC   HL                │
B835 10FA             DJNZ  BKCLP             │
B837 C9               RET                     ┘

B838 D9       P2ALL:  EXX                     ┐
B839 1184DD           LD    DE,0DD84H         │
B83C D9               EXX                     │
B83D 2A80B7           LD    HL,(PATAD)        │
B840 0E10             LD    C,16              │
```

```
B842 0604      P2AL0:  LD      B,4
B844 1A        P2AL1:  LD      A,(DE)
B845 A6                AND     (HL)
B846 13                INC     DE
B847 23                INC     HL
B848 D9                EXX
B849 B6                OR      (HL)
B84A 12                LD      (DE),A
B84B 23                INC     HL
B84C 13                INC     DE
B84D D9                EXX                  ・プレーン別に背景＋くり抜き＋表示を行う
B84E 10F4              DJNZ    P2AL1
B850 D9                EXX
B851 EB                EX      DE,HL
B852 09                ADD     HL,BC
B853 EB                EX      DE,HL
B854 D9                EXX
B855 0D                DEC     C
B856 20EA              JR      NZ,P2AL0
B858 C9                RET

B859 2100BD    PUTP3:  LD      HL,BKDAT
B85C D35C              OUT     (5CH),A
B85E CDA6B8            CALL    DIFE8
B861 D35D              OUT     (5DH),A      ・G.VRAMの余りに背景を作成
B863 CDA6B8            CALL    DIFE8
B866 D35E              OUT     (5EH),A
B868 CDA6B8            CALL    DIFE8
B86B D35F              OUT     (5FH),A
B86D DB32              IN      A,(32H)
B86F CBF7              SET     6,A
B871 D332              OUT     (32H),A
B873 3E80              LD      A,10000000B
B875 D335              OUT     (35H),A
B877 2A80B7            LD      HL,(PATAD)
B87A AF                XOR     A            ・G.VRAMの余りにピーヨを重ね合わせる
B87B CDA4B8            CALL    DIF34
B87E 3E67              LD      A,01100111B
B880 CDA4B8            CALL    DIF34
B883 3E57              LD      A,01010111B
B885 CDA4B8            CALL    DIF34
B888 3E37              LD      A,00110111B
B88A CDA4B8            CALL    DIF34
B88D 3E90              LD      A,10010000B
B88F D335              OUT     (35H),A
B891 2180FE            LD      HL,0FE80H    ・G.VRAMの余りから指定位置にピーヨを転送
B894 118CDD            LD      DE,0DD8CH
B897 CDF4B7            CALL    PUTCC
B89A AF                XOR     A            ・メインメモリをセレクト
B89B D335              OUT     (35H),A
B89D DB32              IN      A,(32H)
B89F CBB7              RES     6,A          ・独立アクセスモードにしてリターン
B8A1 D332              OUT     (32H),A
B8A3 C9                RET
B8A4 D334      DIF34:  OUT     (34H),A      ・ALUのモード設定
B8A6 1180FE    DIFE8:  LD      DE,0FE80H
B8A9 EDA0 *40          LDI     4*16         ・FE80Hへ64バイト転送
B929 C9                RET
```

ピーヨの表示方法は、画面左側から順に次のようになっていますが、おそらくその違いを画面から直接判別することはできないでしょう。

①背景消去 → ピーヨ表示（ALU使用）：面アクセス回数＝7回
②背景＋ピーヨ同時表示（独立アクセスモード）：面アクセス回数＝3回
③グラフィックVRAMのFE80H以降へ、①と同じ方法（アドレスは連続）でデータ作成 → 表示位置へデータを3面同時転送（ALU使用）：面アクセス回数＝8回

このプログラムによる実験では、消去用背景とキャラクタのサイズが同一かつ小さいので、目に見える違いはありません。しかし、種類があれば優劣がつくのは世の常です。一見すると②が最も効率が良さそうですが、データ作成に要する時間が想像以上にかかっています。そこで、最終的にグラフィックVRAM 3面の横1列（4＊3バイト）を完成させるのに要するステート数を、プログラムから計算し比較してみましょう。

なお、グラフィックVRAMの次ラインへの加算時間（アドレス計算＋カウンタ処理）は含みますが、バンク切り換えやモード切り換えに要する時間などその他の細かい部分は無視しています。

	①	②	③	
1ライン処理 次ライン計算	64*7 40*7	294*3 43*3	64*8 40*1	←アドレスの上位変化はないものとする
トータル	728	1011	552	

一番効率の悪そうな③の方法が、意外や一番速度効率がいいという結果です。このことは、理想的なアルゴリズムが必ずしもプログラムとしてはベストではないことを裏付けています。本来、アルゴリズムもプログラムも『Simple is best』なのですが、両立することが難しい場合はプログラムをシンプルにしたほうがいいということです。

プログラム中、②の前後では透明データを反転させていますが、これはピーヨのデータがALU用であるためです。実際に②を利用するのであれば、その方式に合ったデータを用意するのが当然ですから、この部分は比較の対象から外してください。

最後に、このプログラムに含まれている細かなテクニックについて述べておきます。まず、プログラムの先頭部分で左右反転したキャラクタ・データを作成していますが、左右反転というのはデータを左右反転するだけでなく、その並びをも入れ替えなければなりません（図2-5-3）。

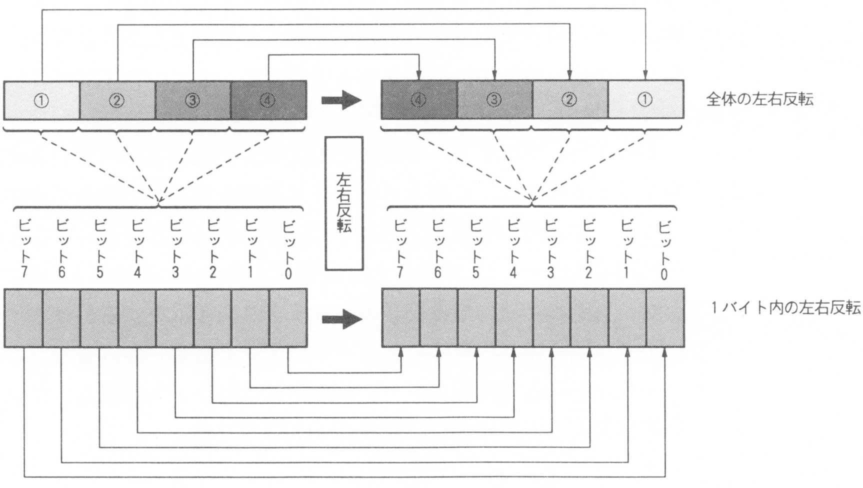


図2-5-3　左右反転データの作成

1バイトのデータを左右反転する命令はありませんから、プログラムで左右反転データを作ることになります（ラベルREVRSの部分)。これはプログラムは簡単でも、8回もループしますから結構実行時間のかかる作業です（1バイトにつき正味258ステート)。今回は最初に左右反転キャラクタ・データを作成しましたが、もしもデータを左右反転しながら表示するという場合、これでは速度的に無理があります。そこで、1バイトを迅速に左右反転するテクニックを紹介しましょう。

●最初に「MKRVD」をコール

```
REVDA:    EQU     xx00H

MKRVD:    LD      HL, REVDA
          XOR     A
REVL0:    LD      B, 8
REVL1:    RRCA
          RL      (HL)
          DJNZ    REVL1
          INC     HL
          INC     A
          JR      NZ, REVL0
          RET
```

●利用時（26ステートでOK）

```
          :
          LD      (GETRV+1), A
GETRV:    LD      A, (REVDA)
          :
```

まず、0～FFHの左右反転データをメモリ上に作成します。このとき、データを作成するメモリの先頭アドレスは、下位8ビットが必ず‘00’になるようにします。これが、たったの26ステートで左右反転データを得るためのポイントです。つまり、「左右反転データのある下位アドレス＝左右反転したいデータ」となりますから、使用時の例に示されているように簡単に左右反転データが取り出せるのです。

従来の左右反転（258ステート）に比べ1／10のステート数ですから、速度的にはデータを左右反転しながら表示することも十分に可能です。ただし、左右反転は横1列の並びも反転（表示アドレスの逆行）させなければならず、状況によっては速度的な負担が大きすぎることもあるかもしれません。このあたりの微妙な判断は、その時々で臨機応変に対応するしかないでしょう。なお、左右反転データがバンク切り換え中に利用される場合、データエリア（REVDA）はC000H以前でなければなりません。

もう1つ、プログラムの最後でLDI命令を64個も並べていますが、本当に速度を追求する場合は、このようにメモリ消費を無視して命令を連続させることも大切です。ここだけを取り上げればLDIR命令でもどうということはありませんが(差は245ステート)、キャラクタ表示完了時には、差はこの7倍にもなります。しかも、実際のゲームでは複数のキャラクタを表示をするわけですから、こういった小さな積み重ねが全体のスピードに大きく影響を与えるのです。

メモリ節約と速度追求……。こういったバランス感覚も、マシン語ゲームにおいては欠くことのできないテクニックなのです。

第3章　画面処理とアイデア

よく『アイデアは無限』といいますが、PC-8801シリーズのゲームにも実に多くのアイデアが隠されています。プログラミングに少しでも興味があるならば、ゲームを表現している画面処理のアイデアも見逃すわけにはいきません。ところで……。

人間には持って生まれた天分というものがあります。これは、ときには才能ともいわれますが、要するに自己を表現するのに最も適した手段という意味です。それが若くして発見できれば天才となり、一生発見できなければ凡人となるのです。音楽家、漫画家、スポーツ選手……ジャンルを問わず自分の世界を表現できる人は輝いています。もちろん、それは天分に「努力」という磨きをかけた結果であることは言うまでもありません。

とはいえ、それもこれも自分の天分を発見してからのお話です。もっと簡単に自分の天分を見極めることができれば、苦労も挫折も少なくなるのですが……。

実は、PC-8801シリーズという機種は「プログラミングによる自己表現」の天分発見機なのです。1枚のテキスト画面とカラーグラフィック画面……。たったこれだけで、どこまで他人と違った自分の世界を表現することができるか、これはもはやマニュアルにはないアイデアの世界です。

もし、PC-8801シリーズでオリジナルな画面処理ができるようになれば、それは「あなたの天分が発見された」ということかもしれません。



3-1　テキスト画面マスク

日常、何気なく使っていたものが、ある日突然貴重なものであることがわかり、一躍脚光を浴びることがあります。例えば、水不足になって初めて水はその価値を認められたわけですが、そうなると「湯水のごとく使う」という言葉は死語同然です。一方、オイルショックで石油の価値が見直されましたが、それまで意識せずに使っていた「油断」という言葉が、それとは逆に本来の意味で急浮上したのです。とかく、モノの価値というのは流動的です。石油に代わる新エネルギー源が発見されれば、「油断」だっていつ死語になるかもしれません。まさに油断大敵なのです。

従来（PC-8801mkⅡ以前)、テキスト画面というのはグラフィック画面を利用するゲームにとって、表示の邪魔であるだけでなく実行速度も遅くする‘お荷物’に近い存在でした（「1-3　テキスト画面の豆知識」参照)。しかし、PC-8801SR以降はDMAによる速度低下がなくなり、テキスト画面の存在価値はにわかに高まりました。

といっても、テキスト画面自体の能力が高くなったわけではありません。調子に乗ってグラフィック画面のように存在をアピールしたのではボロが出てしまいます。テキスト画面は自らの存在を隠すことで、主役であるグラフィック画面をより一層引き立てるのが役目なのです。自分を隠してグラフィック画面も隠す……。つまり、グラフィック画面の部分マスクがテキスト画面の存在価値なのです。

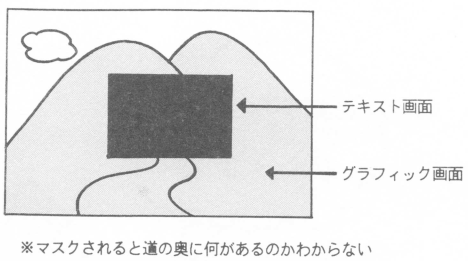


図3-1-1　テキスト画面によるマスク

テキスト画面によるグラフィック画面マスクには、アイデア次第でいろいろな利用法が考えられます。基本的には見せたくないものにフタをするということですが、それだけにフタの存在を知られないようにするテクニックも要求されるわけです。ここでは、そういった隠すテクニックを隠さずに紹介していきます。

(1) ブラインド効果

グラフィック画面の一部（または全部）を描き換えるとき、描く作業を見せたくない場合があります。そこで、テキスト画面でマスクをするわけですが、いき

なり黒マスクをするのではなく、ブラインドが開閉するようにオシャレをしようというのです。

```
WIDTH 80,25
CONSOLE 0,25,0,1
CLEAR,&HB5FF:A=&HB700:CLS:CALL A
```

LIST 3.1

```
                         ORG     0B700H

B700 F3                  DI                    ・割込み禁止
B701 21C8F3              LD      HL,0F3C8H    ┐
B704 112800              LD      DE,120-80    │
B707 AF                  XOR     A            │
B708 0E19                LD      C,25         │
B70A 0650       TEXL0:   LD      B,80         │
B70C 77         TEXL1:   LD      (HL),A       ├・テキスト画面を0でクリア
B70D 23                  INC     HL           │
B70E 10FC                DJNZ    TEXL1        │
B710 19                  ADD     HL,DE        │
B711 0D                  DEC     C            │
B712 20F6                JR      NZ,TEXL0     ┘
B714 2100C0              LD      HL,0C000H    ┐
B717 1101C0              LD      DE,0C001H    │
B71A 017F3E              LD      BC,80*200-1  │
B71D D35C                OUT     (5CH),A      ├・B面（グラフィック）をFFHで埋める
B71F 36FF                LD      (HL),0FFH    │
B721 EDB0                LDIR                 │
B723 D35F                OUT     (5FH),A      ┘
B725 DD2100B7            LD      IX,0B700H    ┐
B729 CDC3B7              CALL    CCBOX        ┘・タイリング・ボックスを描く
B72C DB00       DLOOP:   IN      A,(0)        ┐
B72E 0F                  RRCA                 │
B72F 0F                  RRCA                 ├・[1] が押されていればTCLS1へ
B730 300D                JR      NC,TCLS1     ┘
B732 0F                  RRCA                 ┐
B733 3011                JR      NC,TCLS2     ┘・[2] が押されていればTCLS2へ
B735 0F                  RRCA                 ┐
B736 3020                JR      NC,TCLS3     ┘・[3] が押されていればTCLS3へ
B738 DB09                IN      A,(9)        ┐
B73A 07                  RLCA                 ├・[ESC] が押されていなければDLOOPへ
B73B 38EF                JR      C,DLOOP      ┘
B73D FB                  EI                    ・割込み許可
B73E C9                  RET                   ・BASICへ戻る

B73F 2107B8     TCLS1:   LD      HL,CLDT1     ┐
B742 3E08                LD      A,00001000B  ├・CLDT1で開閉中にタイリング・ボックスのカラー
B744 1805                JR      TVCC9        ┘  を変更

B746 210FB8     TCLS2:   LD      HL,CLDT2     ┐
B749 3E08                LD      A,00001000B  │
B74B CD7FB7     TVCC9:   CALL    ATRCC        │
B74E CD8EB7              CALL    TVCS9        ├・CLDT2で開閉中にタイリング・ボックスのカラー
B751 3EE8                LD      A,11101000B  │  を変更
B753 CD7FB7              CALL    ATRCC        │
B756 18D4                JR      DLOOP        ┘

B758 2117B8     TCLS3:   LD      HL,CLDT3     ┐
B75B 3E18                LD      A,00011000B  │
B75D CD7FB7              CALL    ATRCC        │
B760 0608                LD      B,8          │
B762 CDA6B7     TC3L0:   CALL    TVCLS        │
B765 23                  INC     HL           │
B766 10FA                DJNZ    TC3L0        │
B768 CDC3B7              CALL    CCBOX        │
```

```
B76B 2117B8            LD    HL,CLDT3          ・CLDT3で開閉中にタイリング・ボックスのカラー
B76E 0608              LD    B,8                を変更
B770 7E       TC3L1:   LD    A,(HL)
B771 23                INC   HL
B772 2F                CPL
B773 CDA7B7            CALL  TVACL
B776 10F8              DJNZ  TC3L1
B778 3EE8              LD    A,11101000B
B77A CD7FB7            CALL  ATRCC
B77D 18AD              JR    DLOOP

B77F E5       ATRCC:   PUSH  HL
B780 2119F4            LD    HL,0F419H
B783 117800            LD    DE,120
B786 0619              LD    B,25
B788 77       ATRCL:   LD    (HL),A            ・アトリビュートをAレジスタの指定で変更する
B789 19                ADD   HL,DE
B78A 10FC              DJNZ  ATRCL
B78C E1                POP   HL
B78D C9                RET

B78E 0608     TVCS9:   LD    B,8
B790 CDA6B7   TVXL0:   CALL  TVCLS             ・タイリング・ボックス部分を徐々に閉じる
B793 23                INC   HL
B794 10FA              DJNZ  TVXL0
B796 E5                PUSH  HL
B797 CDC3B7            CALL  CCBOX             ・タイリング・ボックスのカラーを変更する
B79A E1                POP   HL
B79B 0608              LD    B,8
B79D 2B       TVXL1:   DEC   HL
B79E CDA6B7            CALL  TVCLS             ・タイリング・ボックス部分を徐々に開ける
B7A1 10FA              DJNZ  TVXL1
B7A3 AF                XOR   A
B7A4 1801              JR    TVACL

B7A6 7E       TVCLS:   LD    A,(HL)
B7A7 E5       TVACL:   PUSH  HL
B7A8 C5                PUSH  BC
B7A9 2142F4            LD    HL,0F442H
B7AC 114600            LD    DE,120-50
B7AF 0E14              LD    C,20
B7B1 0632     TVCL0:   LD    B,50
B7B3 77       TVCL1:   LD    (HL),A
B7B4 23                INC   HL
B7B5 10FC              DJNZ  TVCL1             ・タイリング・ボックス部分を指定データで埋め、
B7B7 19                ADD   HL,DE              ウェイトを取る
B7B8 0D                DEC   C
B7B9 20F6              JR    NZ,TVCL0
B7BB 0602              LD    B,2
B7BD CDE1B7            CALL  VWAIT
B7C0 C1                POP   BC
B7C1 E1                POP   HL
B7C2 C9                RET

B7C3 D35C     CCBOX:   OUT   (5CH),A
B7C5 DD7E00            LD    A,(IX+0)
B7C8 CDF0B7            CALL  CSBOX
B7CB D35D              OUT   (5DH),A
B7CD DD7E01            LD    A,(IX+1)
B7D0 CDF0B7            CALL  CSBOX             ・タイリング・ボックスのカラーを変更（IXの中身を
B7D3 D35E              OUT   (5EH),A            乱数とする）
B7D5 DD7E02            LD    A,(IX+2)
B7D8 CDF0B7            CALL  CSBOX
B7DB D35F              OUT   (5FH),A
B7DD DD23              INC   IX
B7DF 0608              LD    B,8
B7E1 DB40     VWAIT:   IN    A,(40H)
B7E3 E620              AND   00100000B
B7E5 20FA              JR    NZ,VWAIT
B7E7 DB40     VWAT2:   IN    A,(40H)           ・ウェイト
```

```
B7E9 E620                AND     00100000B
B7EB 28FA                JR      Z,VWAT2
B7ED 10F2                DJNZ    VWAIT
B7EF C9                  RET

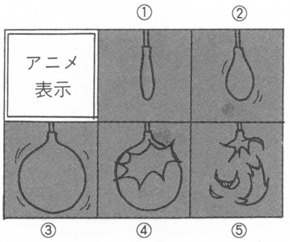
B7F0 2182C2     CSBOX:   LD      HL,0C000H+80*8+2
B7F3 111E00              LD      DE,80-50
B7F6 0EA0                LD      C,20*8
B7F8 0632       CSBL0:   LD      B,50
B7FA 77         CSBL1:   LD      (HL),A
B7FB 0F                  RRCA
B7FC 23                  INC     HL
B7FD 10FB                DJNZ    CSBL1               ・プレーン別にタイリング・ボックスのカラー
B7FF 0F                  RRCA                          を変更
B800 0F                  RRCA
B801 0F                  RRCA
B802 19                  ADD     HL,DE
B803 0D                  DEC     C
B804 20F2                JR      NZ,CSBL0
B806 C9                  RET

B807 80818283   CLDT1:   DB      '▁▂▃▄▅▆▇█'          ・テキスト開閉データ－1
B80B 84858687
B80F 88898A8B   CLDT2:   DB      '▏▎▍▌▋▊▉█'          ・テキスト開閉データ－2
B813 8C8D8E87
B817 0103070F   CLDT3:   DB      1,11B,111B,1111B,10001111B  ・テキスト開閉データ－3
B81B 8F
B81C CFEFFF              DB      11001111B,11101111B,0FFH
```

プログラム（LIST 3.1）を実行すると、青い画面の一部分に妙なタイリングで塗られたボックスが現れます。ここで[1]～[3]キーのどれかを押すと、ボックスの部分が異なったタイプのブラインドで開閉します。ここでは、閉じている間にボックスのタイリングを変化させていますが、実際のゲームではこの間に絵を描くわけです。ブラインドの形状を変えたり、開閉パターンを変化させる（放射線状にする等）ことで、また違った雰囲気を出すことができます。いろいろと試してみてください。

（2）アニメーション用マスク（PC-8801SR以降対応）

パソコンでアニメーションをするということは、結局グラフィック画面を描き変えるということです。当然、そのためにはデータが必要です。データはメインメモリに置くこともできますが、グラフィック・データは量が多いのでメモリの確保が大変です。そこで、グラフィックVRAMの一部に絵としてデータを置き、その部分は見えないようにテキスト画面でマスクをします。そして、拡張グラフィックス機能（3面同時転送）を使って転送すれば、アッという間に作画完了です。複数のパターンを用意すれば、簡単にアニメーションができます（図3-1-2)。



・アニメ表示部以外は、パターンを描いてマスクする
・①～⑤の順でアニメ表示部へ転送すると、アニメーションとなる

図3-1-2　テキスト画面マスクと転送によるアニメーション

この方法（テキスト画面マスクと3面同時転送）は、アニメーションに限らず速度重視の画面処理には大変有効です。最近では、このテクニックゆえに成立しているゲームも少なくありません。もしゲーム画面がフル画面（640×200ドット）でなかったり、あるいは画面のどこかに黒い部分があるという場合は、まず間違いなく見えない部分はデータだらけと判断していいでしょう。

次のプログラムは、このテクニックにその他の拡張グラフィックス機能をミックスした重ね合わせ処理です。2つのパターンが上下に移動するという平凡な内容ですが、ぜひ非凡な応用法を考えてください。実行方法はマシン語プログラム（LIST 3.2）をロードした後、BASICプログラム（LIST 3.3）を走らせます。

LIST 3.2

```
                         ORG    0B700H

B700 F3                  DI                              ・割込み禁止
B701 3E08                LD     A,00001000B              ・アトリビュートの設定
B703 CD9AB7              CALL   ATSET
B706 21C8F3              LD     HL,0F3C8H
B709 112800              LD     DE,120-80
B70C 3E87                LD     A,'■'
B70E 011950              LD     BC,80*100H+25
B711 CDA7B7              CALL   TVBOX
B714 3E20                LD     A,' '                    ・テキスト画面マスクセット
B716 21E2F3              LD     HL,0F3C8H+26
B719 115C00              LD     DE,120-28
B71C 01191C              LD     BC,28*100H+25
B71F CDA7B7              CALL   TVBOX
B722 DB32                IN     A,(32H)
B724 CBF7                SET    6,A                      ・拡張アクセスモード
B726 D332                OUT    (32H),A
B728 3E90                LD     A,10010000B              ・3面同時リード／ライトモード
B72A D335                OUT    (35H),A
B72C 2100C0     TLOOP:   LD     HL,0C000H
B72F 1136C0     MOAD1:   LD     DE,0C000H+54             ・左上のサークルを右側に転送
B732 3E64                LD     A,100
B734 CDBCB7              CALL   LDI26
B737 2140DF              LD     HL,0C000H+80*100
B73A 1176DF     MOAD2:   LD     DE,0C000H+80*100+54
B73D 3E83                LD     A,10000011B
B73F D335                OUT    (35H),A                  ・左下のカラー3のデータを右側に
B741 AF                  XOR    A                          ビットリセット（黒）で転送
B742 D334                OUT    (34H),A
B744 CDB2B7              CALL   CLD26
B747 3E86                LD     A,10000110B
B749 D335                OUT    (35H),A
B74B 3E06                LD     A,6                      ・左下のカラー6のデータを右側にそのまま転送
B74D D334                OUT    (34H),A
B74F CDB2B7              CALL   CLD26
B752 3E90                LD     A,10010000B
```

```
B754 D335                OUT    (35H),A
B756 2136C0              LD     HL,0C000H+54        ・右側に合成されたサークルを中央に転送
B759 111BC0              LD     DE,0C000H+27
B75C 3EC8                LD     A,200
B75E CDBCB7              CALL   LDI26
B761 115000      ADNEX:  LD     DE,80
B764 3E64        KAISU:  LD     A,100
B766 3D                  DEC    A
B767 2009                JR     NZ,ST100
B769 CDFBB7              CALL   DENEG
B76C ED5362B7            LD     (ADNEX+1),DE
B770 3E64                LD     A,100
B772 3265B7      ST100:  LD     (KAISU+1),A         ・サークルを上下に動かすための各データを変更
B775 2A30B7              LD     HL,(MOAD1+1)
B778 19                  ADD    HL,DE
B779 2230B7              LD     (MOAD1+1),HL
B77C 2A3BB7              LD     HL,(MOAD2+1)
B77F CDFBB7              CALL   DENEG
B782 19                  ADD    HL,DE
B783 223BB7              LD     (MOAD2+1),HL
B786 DB09                IN     A,(9)
B788 07                  RLCA                       ・[ESC] が押されていなければTLOOPへ
B789 DA2CB7              JP     C,TLOOP
B78C AF                  XOR    A                   ・メインメモリをセレクト
B78D D335                OUT    (35H),A
B78F CD9AB7              CALL   ATSET               ・アトリビュートを戻す
B792 DB32                IN     A,(32H)
B794 CBB7                RES    6,A                 ・独立アクセスモード
B796 D332                OUT    (32H),A
B798 FB                  EI                         ・割込み許可
B799 C9                  RET                        ・BASICへ戻る

B79A 2119F4      ATSET:  LD     HL,0F419H
B79D 117800              LD     DE,120
B7A0 0619                LD     B,25
B7A2 77          ATRCL:  LD     (HL),A              ・アトリビュートを指定値にセット
B7A3 19                  ADD    HL,DE
B7A4 10FC                DJNZ   ATRCL
B7A6 C9                  RET

B7A7 C5          TVBOX:  PUSH   BC
B7A8 77          TBOXL:  LD     (HL),A
B7A9 23                  INC    HL
B7AA 10FC                DJNZ   TBOXL
B7AC C1                  POP    BC                  ・テキスト画面に指定値でボックス作成
B7AD 19                  ADD    HL,DE
B7AE 0D                  DEC    C
B7AF 20F6                JR     NZ,TVBOX
B7B1 C9                  RET

B7B2 E5          CLD26:  PUSH   HL
B7B3 D5                  PUSH   DE
B7B4 3E64                LD     A,100
B7B6 CDBCB7              CALL   LDI26               ・LDI26を100ライン分コール
B7B9 D1                  POP    DE
B7BA E1                  POP    HL
B7BB C9                  RET

B7BC EDA0 *1A    LDI26:  LDI    26
B7F0 013600              LD     BC,80-26
B7F3 09                  ADD    HL,BC
B7F4 EB                  EX     DE,HL
B7F5 09                  ADD    HL,BC               ・Aの値だけLDI 26を繰り返す
B7F6 EB                  EX     DE,HL
B7F7 3D                  DEC    A
B7F8 20C2                JR     NZ,LDI26
B7FA C9                  RET

B7FB 1B          DENEG:  DEC    DE
B7FC 7B                  LD     A,E
```

```
B7FD 2F        CPL              ┐
B7FE 5F        LD     E,A       │
B7FF 7A        LD     A,D       ├・DEの補数を取る
B800 2F        CPL              │
B801 57        LD     D,A       │
B802 C9        RET              ┘
```

LIST 3.3

```
100 'L=3-3
110 WIDTH 80,25
120 CONSOLE 0,25,0,1:CLS 2
130 FOR N=1 TO 7:CIRCLE(104,49),34+N*8,N:NEXT
140 PAINT(104,49),6,1
150 CIRCLE(104,153),60,3:CIRCLE(104,153),90,3
160 PAINT(104,110),6,3
170 CLEAR,&HB5FF:A=&HB700:CALL A:CLS
```

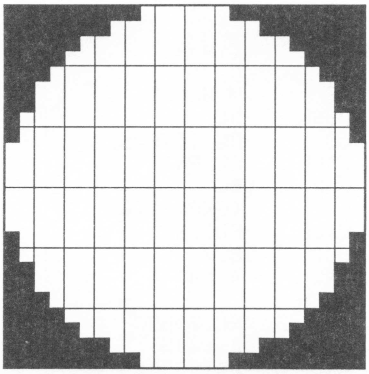
メインのマスク処理は簡単ですが、拡張グラフィックスは選択されたモードと役割をよく理解することが大切です。特に、左下のパターンはカラー0＝透明カラー（下の色が出る）とし、カラー3が黒になるようにしています。また、表示カラーは6だけになっていることにも注意してください。もし、残るすべてのカラーを表示したいという場合は、拡張グラフィックスではカラー単位でしかデータを読めませんから、独立アクセスモードで1面ずつORを取りながら重ね合わせをしなければなりません。

それから、ここでは上のパターンの上下1ラインと下のパターンのフチ取り（カラー3）は、前のパターンの消去用としても利用しています。パターンサイズをフルに使用したり移動ドット数を変更する場合は、右側で合成する前に古いパターンを消去しなければなりません。応用の際には、このあたりにも気を配ってください。

（3）マスク移動

テキスト画面のマスクだからといって、何も固定して使わなければならないということはありません。テキスト画面マスクを動かしてグラフィック画面のほうを固定しても構わないし、あるいは両方動かしてもいいわけです。重要なのは、テキスト画面マスクの存在を隠すことではなく、いかにして画面を効果的に見せるかなのです。

例えば、テキスト・マスクによるスポット処理なんていうのは、非常に簡単で効果的なものです。欠点は、テキスト・グラフィックは160×100ドットと粗いので、円がゴツゴツしたものになってしまうことです（図3-1-3）。



※1ドット（■）はグラフィック画面（640×200）で表すと
4×2ドットに相当する。

図3-1-3　テキスト・グラフィックでの円

しかし、その粗さが目立つのはグラフィック画面がベタで塗られているときで、黒が多ければほとんどわからなくなります。例えば、マシン語プログラム（LIST 3.4）をロードしてから、BASICプログラム（LIST 3.5）を実行してみてください。

LIST 3.4

```
                          ORG     0B700H

B700 F3                   DI                         ・割込み禁止
B701 3E18                 LD      A,00011000B        ┐
B703 CD7FB7               CALL    ATSET              ┘・アトリビュートの設定
B706 21C8F3               LD      HL,0F3C8H          ┐
B709 112800               LD      DE,120-80          │
B70C 3EFF                 LD      A,0FFH             │
B70E 0E19                 LD      C,25               │
B710 0650       TBXL0:    LD      B,80               │
B712 77         TBXL1:    LD      (HL),A             ├・テキスト画面マスク
B713 23                   INC     HL                 │
B714 10FC                 DJNZ    TBXL1              │
B716 19                   ADD     HL,DE              │
B717 0D                   DEC     C                  │
B718 20F6                 JR      NZ,TBXL0           ┘
B71A ED4B8CB7             LD      BC,(TXPOS)         ┐
B71E CDA5B7               CALL    PUTEN              ┘・指定位置にスポットを表示する
B721 ED4B8CB7   MLOOP:    LD      BC,(TXPOS)         ┐
B725 DB00                 IN      A,(0)              │
B727 CB57                 BIT     2,A                │
B729 2008                 JR      NZ,MLP01           │
B72B 0C                   INC     C                  │
B72C 79                   LD      A,C                │
B72D FE27                 CP      39                 │
B72F 28F0                 JR      Z,MLOOP            │
B731 181F                 JR      CPTEN              │
B733 CB67       MLP01:    BIT     4,A                │
B735 2006                 JR      NZ,MLP02           │
B737 05                   DEC     B                  ├・[2] [4] [6] [8] により、エンドチェックをしなが
B738 FA21B7               JP      M,MLOOP            │  らスポットの座標を動かす
B73B 1815                 JR      CPTEN              │
B73D CB77       MLP02:    BIT     6,A                │
B73F 2008                 JR      NZ,MLP03           │
B741 04                   INC     B                  │
B742 78                   LD      A,B                │
B743 FE45                 CP      69                 │
B745 28DA                 JR      Z,MLOOP            │
B747 1809                 JR      CPTEN              │
```

```
B749 DB01      MLP03:  IN    A,(1)
B74B 0F                RRCA
B74C 3814              JR    C,MWAIT
B74E 0D                DEC   C
B74F FA21B7            JP    M,MLOOP
B752 C5        CPTEN:  PUSH  BC
B753 ED4B8CB7          LD    BC,(TXPOS)
B757 CD8EB7            CALL  CLSEN          ・旧スポットをマスクし、新しいスポットを表示する
B75A C1                POP   BC
B75B ED438CB7          LD    (TXPOS),BC
B75F CDA5B7            CALL  PUTEN
B762 0602      MWAIT:  LD    B,2
B764 DB09      WAITL:  IN    A,(9)
B766 07                RLCA
B767 3010              JR    NC,TOBSC
B769 DB40      VWAIT:  IN    A,(40H)
B76B E620              AND   00100000B      ・ウェイトを取ってMLOOPへ（[ESC] が押されて
B76D 20FA              JR    NZ,VWAIT         いればTOBSCへ）
B76F DB40      VWAT2:  IN    A,(40H)
B771 E620              AND   00100000B
B773 28FA              JR    Z,VWAT2
B775 10ED              DJNZ  WAITL
B777 18A8              JR    MLOOP
B779 AF        TOBSC:  XOR   A              ・アトリビュートを元に戻す
B77A CD7FB7            CALL  ATSET
B77D FB                EI                   ・割込み許可
B77E C9                RET                  ・BASICへ戻る

B77F 2119F4    ATSET:  LD    HL,0F419H
B782 117800            LD    DE,120
B785 0619              LD    B,25
B787 77        ATRCL:  LD    (HL),A         ・Aレジスタの値でアトリビュートをセットする
B788 19                ADD   HL,DE
B789 10FC              DJNZ  ATRCL
B78B C9                RET

B78C 0000      TXPOS:  DB    0,0            ・スポットの座標（X, Y）

B78E CDCDB7    CLSEN:  CALL  TXADR
B791 116C00            LD    DE,120-12
B794 3EFF              LD    A,0FFH
B796 060C      CENL0:  LD    B,12
B798 DD7700    CENL1:  LD    (IX+0),A
B79B DD23              INC   IX             ・スポットをマスクする
B79D 10F9              DJNZ  CENL1
B79F DD19              ADD   IX,DE
B7A1 0D                DEC   C
B7A2 20F2              JR    NZ,CENL0
B7A4 C9                RET

B7A5 CDCDB7    PUTEN:  CALL  TXADR          ・BCからTVRAM(IX)、ﾃﾞｰﾀｱﾄﾞﾚｽ(HL)、縦列数(C)を求める
B7A8 116C00            LD    DE,120-12
B7AB 0606      PENL0:  LD    B,6
B7AD 7E        PENL1:  LD    A,(HL)
B7AE DD7700            LD    (IX+0),A       ・スポット（一列の左側）表示
B7B1 23                INC   HL
B7B2 DD23              INC   IX
B7B4 10F7              DJNZ  PENL1
B7B6 E5                PUSH  HL
B7B7 0606              LD    B,6
B7B9 2B        PENL2:  DEC   HL
B7BA 7E                LD    A,(HL)
B7BB 07                RLCA
B7BC 07                RLCA                 ・スポット（一列の右側）表示
B7BD 07                RLCA
B7BE 07                RLCA
B7BF DD7700            LD    (IX+0),A
B7C2 DD23              INC   IX
B7C4 10F3              DJNZ  PENL2
B7C6 E1                POP   HL
```

```
B7C7 DD19                ADD     IX,DE              ・縦列分を表示してスポットを完成
B7C9 0D                  DEC     C
B7CA 20DF                JR      NZ,PENL0
B7CC C9                  RET

B7CD 69         TXADR:   LD      L,C                ・BCからTVRAM (IX)、データアドレス (HL)、
B7CE CB3D                SRL     L                    縦列数 (C) を求める
B7D0 2600                LD      H,0
B7D2 54                  LD      D,H
B7D3 5D                  LD      E,L
B7D4 29                  ADD     HL,HL
B7D5 29                  ADD     HL,HL
B7D6 29                  ADD     HL,HL
B7D7 29                  ADD     HL,HL
B7D8 B7                  OR      A
B7D9 ED52                SBC     HL,DE
B7DB 29                  ADD     HL,HL
B7DC 29                  ADD     HL,HL
B7DD 29                  ADD     HL,HL
B7DE 58                  LD      E,B
B7DF 1600                LD      D,0
B7E1 19                  ADD     HL,DE
B7E2 11C8F3              LD      DE,0F3C8H
B7E5 19                  ADD     HL,DE
B7E6 E5                  PUSH    HL
B7E7 DDE1                POP     IX
B7E9 21F6B7              LD      HL,CDAT1
B7EC CB39                SRL     C
B7EE 0E06                LD      C,6
B7F0 D0                  RET     NC
B7F1 211AB8              LD      HL,CDAT2
B7F4 0C                  INC     C
B7F5 C9                  RET

B7F6 FFFF3713   CDAT1:   DB      0FFH,0FFH,37H,13H,1,0  ・C (X座標) =偶数時のスポットデータ
B7FA 0100
B7FC 7F010000            DB      7FH,1,0,0,0,0
B800 0000
B802 01000000            DB      1,0,0,0,0,0
B806 0000
B808 08000000            DB      8,0,0,0,0,0
B80C 0000
B80E EF080000            DB      0EFH,8,0,0,0,0
B812 0000
B814 FFFFCE8C            DB      0FFH,0FFH,0CEH,8CH,8,0
B818 0800

B81A FFFFFF7F   CDAT2:   DB      0FFH,0FFH,0FFH,7FH,37H,33H  ・C (X座標) =奇数時の
B81E 3733                                                      スポットデータ
B820 FF370100            DB      0FFH,37H,1,0,0,0
B824 0000
B826 17000000            DB      17H,0,0,0,0,0
B82A 0000
B82C 00000000            DB      0,0,0,0,0,0
B830 0000
B832 8C000000            DB      8CH,0,0,0,0,0
B836 0000
B838 FFCE0800            DB      0FFH,0CEH,8,0,0,0
B83C 0000
B83E FFFFFFEF            DB      0FFH,0FFH,0FFH,0EFH,0CEH,0CCH
B842 CECC
```

LIST 3.5

```
100 'L=3-5
110 WIDTH 80,25
120 CONSOLE 0,25,0,1:CLS 2
130 FOR N=0 TO 639 STEP 19
140 LINE(N,0)-(639-N,199),(N AND 7)
150 NEXT
160 FOR N=0 TO 199 STEP 9
170 LINE(0,N)-(639,199-N),(N AND 7)
180 NEXT
190 CLEAR,&HB5FF:A=&HB700:CALL A
```

移動時には多少ゴツゴツしますが、停止時にはこのスポットがテキスト画面によるマスクとはまず判断できません。どんなに高度なテクニックも、無条件に使用できるわけではないし、簡単なテクニックで済むものは簡単に済ませるべきです。大切なのは、状況を的確に把握し、それに合わせたテクニックを選択することなのです。

ところで、このプログラム（LIST 3.4）では、メモリ節約のためスポットの表示データは左半分用しか持っていません。右半分は、プログラム（RLCA ×4回）で作っています。テキスト・グラフィックにおける左右反転はどうなっているか、図1-2-3を参考にしながら確認してください。また、ここではスポットを2ドット単位で動かしていますから、Y座標の偶数／奇数でデータを変えなければなりません。これは、テキスト・グラフィックならではの面倒な部分といえるでしょう。

テキスト画面の利用法は、ここに紹介したものがすべてではありません。グラフィック画面を見えなくするだけでなく、表示している自分自身を見えなくしてもいいのです。もちろん、邪魔だから見えなくするのではなく、見せたくないから見えなくするということです。文章ではわかりにくいかもしれませんが、カラー0の文字とグラフィック画面を重ね合わせるとか、オフにしたテキスト画面の内容を参照する……とか、アイデア次第で利用法は無限です。‘ないもの’を利用することはできませんが、せっかく‘あるもの’を利用しない手はありません。テキスト画面を大いに活用しましょう。



3-2 プレーンの分割（その1）

アイデアというと、すぐ「アイデア→特許大発明→大金持ち→自分には無理」と連想してしまいますが、頭のいい人は他人のアイデアを応用して新しいアイデアを考えるそうです。聞くところによると、それが小発明のコツであるとか……。

すでに『マシン語ゲームプログラミング』（アスキー発行）を読まれた方は、『スカイ・ブルーザー』というタイトルのスクロール・シューティング・ゲームをご存じのはずです。あの旧式のゲームは、プレーンを分割して使うことによってスクロールと重ね合わせ処理を成立させていました。知らない方もいるかもしれませんので、もう一度その基本を振り返ってみることにしましょう。

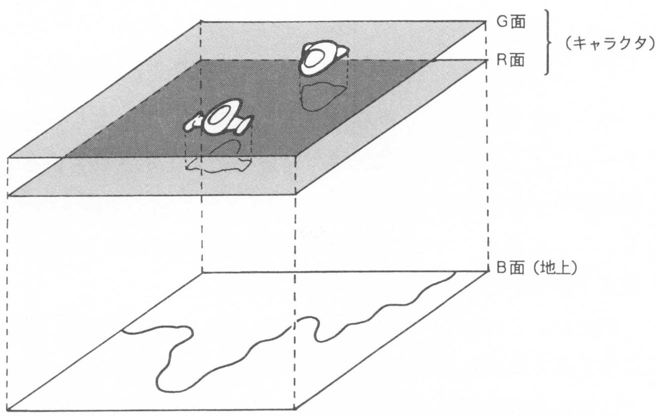


図3-2-1 『スカイ・ブルーザー』の画面処理

図3-2-1を見てください。スクロールする地上背景はB面で、自機および襲来する謎の飛行物体はR／G面を使って表示しています。こうすれば、地上は地上で勝手にスクロールさせることができるし、キャラクタは地上との重ね合わせ処理を気にせずに表示することができます。もちろん、単に使用するプレーンを分けただけでは分割したことにはなりません。このままでは、キャラクタ（R／G面）が地上（B面）の影響を受け色変化を起こしてしまうからです。そこで、R／G面がB面から独立して発色されるよう、パレットを図3-2-2のように変更しています。

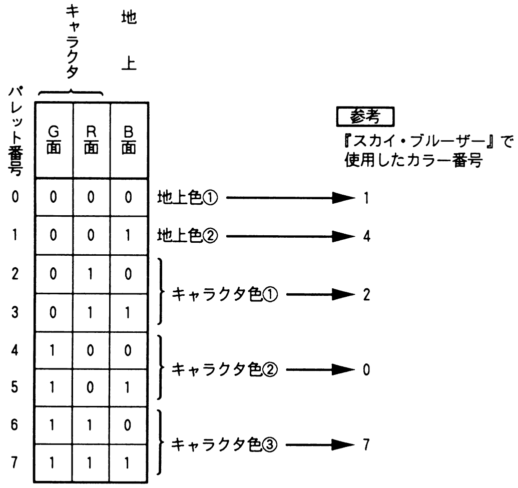


図 3-2-2　パレットによるプレーン分割

キャラクタに使用できる色数は3色になってしまいましたが、これで地上パターンによる影響はなくなります。参考として示されている『スカイ・ブルーザー』用のカラーを変更すれば、ゲームのイメージもある程度は変わるでしょう。しかし、ここにちょっとしたアイデアを加えるとまったく違った効果を得ることができます。

例えば、このプレーン分割ではキャラクタ（R／G面）が地上（B面）より優先して表示されていますが、これは単にゲーム設定によるものです。実際にはB／R／G各面は同格の存在ですから、プレーンの優先順位はパレット変更次第なのです。とはいえ、パレット番号とB／R／Gのビット関係（図3-2-2参照）から、G→R→Bと表示優先順位が低くなるように設定したほうがわかりやすいのは事実です。

次のプログラム（LIST 3.6／LIST 3.7）は、B／R面でキャラクタを表示し、G面で上空を流れる雲を表したものです。まずは、プログラムを実行して雰囲気の違いを味わってください。実行は、マシン語（LIST 3.6）をロードしてからBASIC（LIST 3.7）を走らせます。

LIST 3.6

```
                          ORG     0B700H

B700 F3                   DI                     ・割込み禁止
B701 DB71                 IN      A,(71H)      ┐
B703 F5                   PUSH    AF           │
B704 3EFF                 LD      A,0FFH       │・メインROMセレクト
B706 D371                 OUT     (71H),A      ┘
B708 3AD779               LD      A,(79D7H)    ┐
B70B FE33                 CP      '3'          │
B70D 3806                 JR      C,PALCC      │
B70F DB32                 IN      A,(32H)      │・デジタルカラー・モードにする
B711 E6DF                 AND     11011111B    │
B713 D332                 OUT     (32H),A      ┘
B715 2131B8      PALCC:   LD      HL,PALDT     │
B718 015408               LD      BC,854H      │
```

```
B71B 7E          PALCL:  LD     A,(HL)
B71C ED79                OUT    (C),A            ・パレットを変更
B71E 0C                  INC    C
B71F 23                  INC    HL
B720 10F9                DJNZ   PALCL
B722 CD39B8              CALL   CHRDI            ・キャラクタの表示
B725 D35E        MLOOP:  OUT    (5EH),A
B727 217FFE              LD     HL,0FE7FH
B72A 11CFFE              LD     DE,0FE7FH+80
B72D 06C8                LD     B,200
B72F 0EFF        KUMOL:  LD     C,0FFH
B731 EDA8 *50            LDD    80               ・G面を下にループスクロールさせる
B7D1 05                  DEC    B
B7D2 C22FB7              JP     NZ,KUMOL
B7D5 21CFFE              LD     HL,0FE7FH+80
B7D8 015000              LD     BC,80
B7DB EDB8                LDDR
B7DD D35F                OUT    (5FH),A
B7DF 0602                LD     B,2              ・ウェイトの回数
B7E1 DB09        WAITL:  IN     A,(9)
B7E3 07                  RLCA                    ・[ESC] が押されていればTOBSCへ
B7E4 303C                JR     NC,TOBSC
B7E6 07                  RLCA                    ・[SPACE] が押されていなければUDKUMへ
B7E7 3816                JR     C,UDKUM
B7E9 3E00        UPDWN:  LD     A,0
B7EB B7                  OR     A
B7EC 2023                JR     NZ,VWAIT
B7EE 3C                  INC    A
B7EF 32EAB7              LD     (UPDWN+1),A
B7F2 AF                  XOR    A                ・必要に応じパレットを変更する（5→0、6→2、
B7F3 D359                OUT    (59H),A            7→6）
B7F5 3E02                LD     A,2
B7F7 D35A                OUT    (5AH),A
B7F9 3E06                LD     A,6
B7FB D35B                OUT    (5BH),A
B7FD 1812                JR     VWAIT
B7FF 3AEAB7      UDKUM:  LD     A,(UPDWN+1)
B802 B7                  OR     A
B803 280C                JR     Z,VWAIT
B805 AF                  XOR    A
B806 32EAB7              LD     (UPDWN+1),A      ・必要に応じパレットを変更する（5→7、6→7、
B809 3E07                LD     A,7                7→7）
B80B D359                OUT    (59H),A
B80D D35A                OUT    (5AH),A
B80F D35B                OUT    (5BH),A
B811 DB40        VWAIT:  IN     A,(40H)
B813 E620                AND    00100000B
B815 20FA                JR     NZ,VWAIT
B817 DB40        VWAT2:  IN     A,(40H)          ・ウェイトを取ってMLOOPへ
B819 E620                AND    00100000B
B81B 28FA                JR     Z,VWAT2
B81D 10C2                DJNZ   WAITL
B81F C325B7              JP     MLOOP
B822 AF          TOBSC:  XOR    A
B823 015408              LD     BC,854H
B826 ED79        PALIL:  OUT    (C),A            ・パレット・イニシャライズ
B828 3C                  INC    A
B829 0C                  INC    C
B82A 10FA                DJNZ   PALIL
B82C F1                  POP    AF               ・ROMの状態を元に戻す
B82D D371                OUT    (71H),A
B82F FB                  EI                      ・割込み許可
B830 C9                  RET                     ・BASICへ戻る

B831 05000206    PALDT:  DB     5,0,2,6,7,7,7,7  ・変更用パレットデータ
B835 07070707

B839 2166B8      CHRDI:  LD     HL,CHRDT
B83C D35C                OUT    (5CH),A
B83E CD49B8              CALL   CHLDI
```

```
B841 D35D                   OUT    (5DH),A
B843 CD49B8                 CALL   CHLDI
B846 D35F                   OUT    (5FH),A
B848 C9                     RET
B849 1105EF       CHLDI:    LD     DE,0EF05H
B84C 0EFF                   LD     C,0FFH
B84E 0618                   LD     B,24              ・指定位置にキャラクタを表示
B850 EDA0 *06     CHLLP:    LDI    6
B85C 3E4A                   LD     A,80-6
B85E 83                     ADD    A,E
B85F 3001                   JR     NC,NINCD
B861 14                     INC    D
B862 5F           NINCD:    LD     E,A
B863 10EB                   DJNZ   CHLLP
B865 C9                     RET

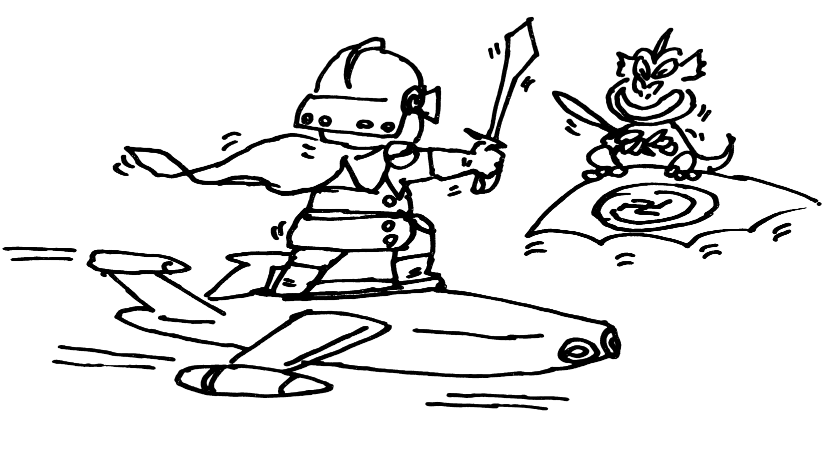
B866 00000FF0     CHRDT:    DB     0,0,0FH,0F0H,0,0,0,0        ・キャラクタ・データ
B86A 00000000
B86E 3FFC0000               DB     3FH,0FCH,0,0,0,0,0FFH,0FFH
B872 0000FFFF
B876 00000003               DB     0,0,0,3,0FDH,77H,0C0H,0
B87A FD77C000
B87E 0007FEFD               DB     0,7,0FEH,0FDH,60H,0,0,0FH
B882 6000000F
B886 FDFEB000               DB     0FDH,0FEH,0B0H,0,0,1FH,0F5H,5FH
B88A 001FF55F
B88E 5800003F               DB     58H,0,0,3FH,0FFH,0FFH,0ACH,0
B892 FFFFAC00
B896 003FFFFF               DB     0,3FH,0FFH,0FFH,5CH,0,0,7FH
B89A 5C00007F
B89E F80FAE00               DB     0F8H,0FH,0AEH,0,0,7FH,0E3H,0A7H
B8A2 007FE3A7
B8A6 D60000FF               DB     0D6H,0,0,0FFH,0CEH,83H,0EBH,0
B8AA CE83EB00
B8AE 00FF9EC1               DB     0,0FFH,09EH,0C1H,0D7H,0,0,0FFH
B8B2 D70000FF
B8B6 BA89EB00               DB     0BAH,89H,0EBH,0,0,0FFH,0BAH,89H
B8BA 00FFBA89
B8BE D70031FF               DB     0D7H,0,31H,0FFH,0B5H,29H,0EBH,8CH
B8C2 B529EB8C
B8C6 6BDFAA51               DB     6BH,0DFH,0AAH,51H,0D1H,0D6H,6FH,0DFH
B8CA D1D66FDF
B8CE D6A3D1F6               DB     0D6H,0A3H,0D1H,0F6H,6FH,0EFH,0F5H,4FH
B8D2 6FEFF54F
B8D6 A3F6FFF7               DB     0A3H,0F6H,0FFH,0F7H,0FFH,0FEH,87H,0FFH
B8DA FFFE87FF
B8DE FFFDFFFE               DB     0FFH,0FDH,0FFH,0FEH,1FH,0FBH,7FH,0FFH
B8E2 1FFB7FFF
B8E6 FFF4FFFE               DB     0FFH,0F4H,0FFH,0FEH,0,0,0FFH,0ABH
B8EA 0000FFAB
B8EE 00000000               DB     0,0,0,0,3FH,0FCH,0,0
B8F2 3FFC0000
B8F6 00000000               DB     0,0,0,0,0,0,0,0
B8FA 00000000
B8FE 07E00000               DB     7,0E0H,0,0,0,0,1FH,0F8H
B902 00001FF8
B906 00000000               DB     0,0,0,0,78H,3EH,0,0
B90A 783E0000
B90E 0001FE7F               DB     0,1,0FEH,7FH,80H,0,0,3
B912 80000003
B916 FCFFC000               DB     0FCH,0FFH,0C0H,0,0,7,0F0H,1FH
B91A 0007F01F
B91E E000000F               DB     0E0H,0,0,0FH,0FFH,0FFH,0F0H,0
B922 FFFFF000
B926 000FF81F               DB     0,0FH,0F8H,1FH,0F0H,0,0,1FH
B92A F000001F
B92E E017F800               DB     0E0H,17H,0F8H,0,0,1FH,0C3H,0ABH
B932 001FC3AB
B936 F800003F               DB     0F8H,0,0,3FH,8EH,85H,0FCH,0
B93A 8E85FC00
```

```
B93E 003F1EC2         DB       0,3FH,1EH,0C2H,0FCH,0,0,3FH
B942 FC00003F
B946 3A82FC00         DB       3AH,82H,0FCH,0,0,3FH,3AH,82H
B94A 003F3A82
B94E FC00007F         DB       0FCH,0,0,7FH,35H,2,0FEH,0
B952 3502FE00
B956 10DF2A02         DB       10H,0DFH,2AH,2,0FFH,8,13H,0DFH
B95A FF0813DF
B95E 9405FDC8         DB       094H,5,0FDH,0C8H,1FH,0EFH,0C5H,43H
B962 1FEFC543
B966 FBF83FF7         DB       0FBH,0F8H,3FH,0F7H,0F0H,0FH,0F7H,0FCH
B96A F00FF7FC
B96E 3FFDFFFF         DB       3FH,0FDH,0FFH,0FFH,0DFH,0FCH,0,0
B972 DFFC0000
B976 FFFF0000         DB       0FFH,0FFH,0,0,0,0,3FH,0FCH
B97A 00003FFC
B97E 00000000         DB       0,0,0,0,0,0,0,0
B982 00000000
```

LIST 3.7

```
100 'L=3-7
110 WIDTH 80,25
115 CONSOLE 0,25,0,1:CLS 2
120 CIRCLE(195,30),130,4,...2
130 PAINT(195,30),4,4
140 CIRCLE(305,50),100,4,...,15
150 PAINT(305,50),4,4
160 CIRCLE(195,30),130,0,...2
170 CIRCLE(420,145),100,4,...2
180 PAINT(420,145),4,4
190 CLEAR,&HB5FF:A=&HB700:CALL A
```

パレットの状態は図3-2-3のようになっていますが、ここでパレット番号0に注目してください。パレット番号0というのは、B／R／Gどの面も発色していない地色のことですから、存在としてはB／R／G各面の下に位置すると考えることができます。したがって、パレット番号0は単独で地色としてもいいし、最も表示優先順位の低いプレーンとコンビにして使うこともできます。ちなみに、『スカイ・ブルーザー』ではB面（地上）と組み合わせ、地上を2色としていました。今回はキャラクタ表示用のB／R面が表示優先順位が低いので、パレット番号0はイメージとしては空あるいは海面なのですが、キャラクタ・カラーの一部としても使用することができます。



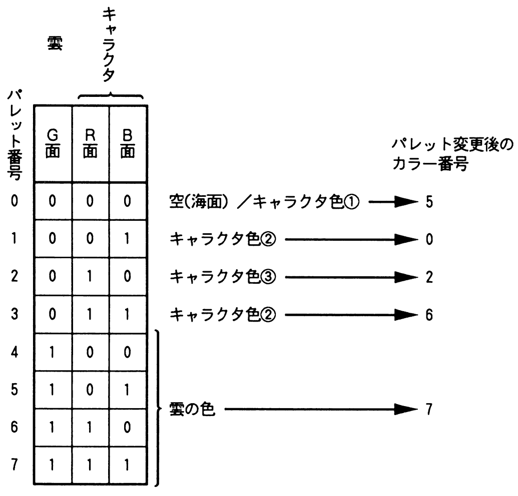


図3-2-3　プレーン分割の変化

もし、パレット番号0をキャラクタに使用しないならば、パレットを5→0、6→2、7→6と変更するだけでキャラクタが雲の上を通過するようになります。ここではキャラクタにパレット番号0を使っているので、その部分だけ雲の影響が出ますが、参考としてスペースキーを押している間だけパレットを変化させています。雲の上を通過するときに受ける影響と、プレーンの表示優先順位の関係を実際に確認してください。

いまにも敵機が現れそうな画面ですが、流れてくるのは同じ形の雲ばかりです。これが本物のゲームなら、雲の形状を変えたり濃霧を起こしたり……と、新たな障害物として楽しむことができるのですが、とりあえずは『スカイ・ブルーザー』との違いを感じられれば目的達成です。というのも、プレーン分割の応用はまだ続編があるからです。

今度は、画面を構成している雲／キャラクタという概念を変えてみましょう。キャラクタというと、どうしてもイメージが小さくなりますが、サイズの限定はどこにもありません。そこで、B／R面（パレット番号0を含んだ4色）を使って、フル画面で1枚の絵を描いてしまおうというのです。

こうなると、G面の役割は必然的にマスクしか残されていません。しかし、そのためにプレーン1枚を使うわけですから、たとえ複雑なタイトルロゴであろうと、形状を問わずにマスクをかけることができます。もちろん、かけたマスクを移動させることも自由自在です。サンプルとして、双眼鏡越しに絵を覗いたプログラムを示しますが、大きな絵の代わりにあの『おばけのピーヨ』を大量に出現させます。何匹出現しているか、バードウォッチングならずにピーヨウォッチングを楽しんでください。もちろん、表示色の制限がピーヨの表示にどう影響を与えたか、データの構造と表示カラーの関係を考えることも忘れずに……。

実行方法は、マシン語（**LIST 3.8**）とピーヨ・データをロード（BLOAD"PIYO"）

してから、BASIC (LIST 3.9) を走らせます (BSAVEされたプログラムにはデータも含まれています)。双眼鏡は、テンキーで上下左右に移動することができます。

LIST 3.8

```
                        ORG     0B700H

B900=           SSDAT:  EQU     0B900H          ・双眼鏡のデータエリア
BF00=           TMEN1:  EQU     0BF00H          ┐
BF40=           BMENS:  EQU     TMEN1+4*16      │・ピーヨのデータエリア
BF80=           RMENS:  EQU     BMENS+4*16      │
BFC0=           GMENS:  EQU     RMENS+4*16      ┘

B700 F3                 DI                      ・割込み禁止
B701 DB71               IN      A,(71H)         ┐
B703 F5                 PUSH    AF              │・メインROMセレクト
B704 3EFF               LD      A,0FFH          │
B706 D371               OUT     (71H),A         ┘
B708 3AD779             LD      A,(79D7H)       ┐
B70B FE33               CP      '3'             │
B70D 3806               JR      C,PALCC         │・デジタルカラー・モードにする
B70F DB32               IN      A,(32H)         │
B711 E6DF               AND     11011111B       │
B713 D332               OUT     (32H),A         ┘
B715 2197B7     PALCC:  LD      HL,PALDT        ┐
B718 015408             LD      BC,854H         │
B71B 7E         PALCL:  LD      A,(HL)          │
B71C ED79               OUT     (C),A           │・パレットを変更
B71E 0C                 INC     C               │
B71F 23                 INC     HL              │
B720 10F9               DJNZ    PALCL           ┘
B722 D35E               OUT     (5EH),A         ┐
B724 2100C0             LD      HL,0C000H       │
B727 1100B9             LD      DE,SSDAT        │
B72A 3E3C               LD      A,15*4          │
B72C 011700     MSSDL:  LD      BC,23           │
B72F EDB0               LDIR                    │・双眼鏡のデータ作成
B731 013900             LD      BC,80-23        │
B734 09                 ADD     HL,BC           │
B735 3D                 DEC     A               │
B736 20F4               JR      NZ,MSSDL        │
B738 D35F               OUT     (5FH),A         ┘
B73A DD219FB7           LD      IX,DIPAD        ┐
B73E 0614               LD      B,20            │
B740 C5         CHRDL:  PUSH    BC              │
B741 DD6E00             LD      L,(IX+0)        │
B744 DD6601             LD      H,(IX+1)        │
B747 CD09B8             CALL    DIPHL           │・全ピーヨを表示する
B74A DD23               INC     IX              │
B74C DD23               INC     IX              │
B74E C1                 POP     BC              │
B74F 10EF               DJNZ    CHRDL           ┘

B751 DB00       MLOOP:  IN      A,(0)           ┐
B753 010001             LD      BC,100H         │
B756 CB57               BIT     2,A             │
B758 286F               JR      Z,MOVES         │
B75A 01FF00             LD      BC,0FFH         │
B75D CB67               BIT     4,A             │
B75F 2868               JR      Z,MOVES         │・[2] [4] [6] [8] のどれかが押されていれば
B761 010100             LD      BC,1            │  MOVESへ
B764 CB77               BIT     6,A             │
B766 2861               JR      Z,MOVES         │
B768 DB01               IN      A,(1)           │
B76A 0100FF             LD      BC,0FF00H       │
B76D 0F                 RRCA                    │
B76E 3059               JR      NC,MOVES        ┘
B770 0602       WAIT:   LD      B,2             ・ウェイト回数
B772 DB09       WAITL:  IN      A,(9)           │
```

```
B774 07                  RLCA                             ・[ESC] が押されていればTOBSCへ
B775 3011                JR      NC,TOBSC
B777 DB40      VWAIT:    IN      A,(40H)
B779 E620                AND     00100000B
B77B 20FA                JR      NZ,VWAIT
B77D DB40      VWAT2:    IN      A,(40H)                  ・ウェイト後MLOOPへ
B77F E620                AND     00100000B
B781 28FA                JR      Z,VWAT2
B783 10ED                DJNZ    WAITL
B785 C351B7              JP      MLOOP
B788 AF        TOBSC:    XOR     A
B789 015408              LD      BC,854H
B78C ED79      PALIL:    OUT     (C),A                    ・パレット・イニシャライズ
B78E 3C                  INC     A
B78F 0C                  INC     C
B790 10FA                DJNZ    PALIL
B792 F1                  POP     AF                       ・ROMの状態を元に戻す
B793 D371                OUT     (71H),A
B795 FB                  EI                               ・割込み許可
B796 C9                  RET                              ・BASICへ戻る

B797 01000706  PALDT:    DB      1,0,7,6,0,0,0,0          ・変更用パレットデータ
B79B 00000000
B79F 00C110C2  DIPAD:    DW      0C100H,0C210H,0C3C2H,0C452H  ・ピーヨ表示位置データ
B7A3 C2C352C4
B7A7 15C930C9            DW      0C915H,0C930H,0CCC3H,0CF09H
B7AB C3CC09CF
B7AF 60D112D3            DW      0D160H,0D312H,0D2E0H,0D640H
B7B3 E0D240D6
B7B7 68D9C2E0            DW      0D968H,0E0C2H,0EE80H,0F040H
B7BB 80EE40F0
B7BF A0F300F5            DW      0F3A0H,0F500H,0F5B0H,0F885H
B7C3 B0F585F8
B7C7 00        SPOSX:    DB      0                        ・双眼鏡X座標
B7C8 00        SPOSY:    DB      0                        ・双眼鏡Y座標

B7C9 3AC7B7    MOVES:    LD      A,(SPOSX)
B7CC 81                  ADD     A,C
B7CD FE3A                CP      80-23+1
B7CF 309F                JR      NC,WAIT
B7D1 32C7B7              LD      (SPOSX),A
B7D4 3AC8B7              LD      A,(SPOSY)                ・双眼鏡の移動後の座標を求める
B7D7 80                  ADD     A,B
B7D8 FE24                CP      50-15+1
B7DA 3094                JR      NC,WAIT
B7DC 32C8B7              LD      (SPOSY),A
B7DF ED4BC7B7  DIPSS:    LD      BC,(SPOSX)
B7E3 04                  INC     B
B7E4 21C0BE              LD      HL,0C000H-80*4
B7E7 114001              LD      DE,80*4
B7EA 19        DIPL0:    ADD     HL,DE
B7EB 10FD                DJNZ    DIPL0
B7ED 09                  ADD     HL,BC
B7EE EB                  EX      DE,HL
B7EF 2100B9              LD      HL,SSDAT
B7F2 3E3C                LD      A,15*4
B7F4 D35E                OUT     (5EH),A                  ・双眼鏡を指定位置に表示
B7F6 011700    DIPL1:    LD      BC,23
B7F9 EDB0                LDIR
B7FB EB                  EX      DE,HL
B7FC 013900              LD      BC,80-23
B7FF 09                  ADD     HL,BC
B800 EB                  EX      DE,HL
B801 3D                  DEC     A
B802 20F2                JR      NZ,DIPL1
B804 D35F                OUT     (5FH),A
B806 C370B7              JP      WAIT                     ・WAITへ

B809 1100BF    DIPHL:    LD      DE,TMEN1
B80C D35C                OUT     (5CH),A
```

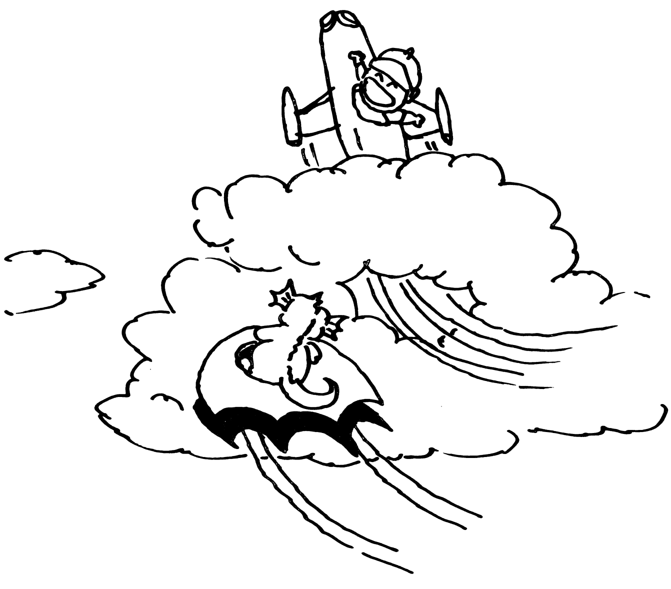
```
B80E CD1CB8           CALL   DIXOR
B811 CD1CB8           CALL   DIXOR
B814 D35D             OUT    (5DH),A
B816 CD1CB8           CALL   DIXOR
B819 D35F             OUT    (5FH),A
B81B C9               RET

B81C E5       DIXOR:  PUSH   HL
B81D 011004           LD     BC,410H
B820 C5       DIXL0:  PUSH   BC
B821 1A       DIXL1:  LD     A,(DE)        ・B/R面とXORを取りながらピーヨを表示
B822 AE               XOR    (HL)
B823 77               LD     (HL),A
B824 23               INC    HL
B825 13               INC    DE
B826 10F9             DJNZ   DIXL1
B828 014C00           LD     BC,80-4
B82B 09               ADD    HL,BC
B82C C1               POP    BC
B82D 0D               DEC    C
B82E 20F0             JR     NZ,DIXL0
B830 E1               POP    HL
B831 C9               RET
```

LIST 3.9

```
110 WIDTH 80,25
120 CONSOLE 0,25,0,1
130 LINE(0,0)-(639,199),4,BF
140 CIRCLE(58,29),50,0
150 PAINT(58,29),0,0
160 CIRCLE(125,29),50,0
170 PAINT(125,29),0,0
180 CLEAR,&HB5FF:A=&HB700:CALL A
```

これまた『スカイ・ブルーザー』とは違った雰囲気ですが、すべて基本は同じアイデアです。これをどう応用しどう発展させるか……、あなたのアイデアがそのカギを握っているのです。



3-3 プレーンの分割(その2)

手品師が「このハンカチにはタネも仕掛けもありません……」と言ってハンカチを胸のポケットから取り出しても、それをそのまま信じる人はいないでしょう。きっと秘密があるハズと、かえってハンカチを注目することになります。しかし、それではますます手品師の思うツボです。なぜなら、観客の目がハンカチに集まれば、もう片方の手はタネも仕掛けも思い通りに扱えるからです。

この場合、ハンカチは観客に見せるための存在であり、本当の手品の主役は見えないところで頑張っているもう片方の手です。とはいえ、それがバレるようでは手品師ではありません。たった1枚のハンカチを右に左に持ちかえては、どちらが主役の手かわからなくなるようにしているのです。

ところで、似たようなことを実行したプログラム(LIST 3.2/LIST 3.3)を覚えていますか。移動するパターンの重ね合わせ処理を、テキスト画面マスクと拡張グラフィックスの3面同時転送機能で実現したプログラムです。見えている部分がハンカチなら、見えない右側の作業はまさに主役の手そのものです。……が、このように役割が明確に分離していたのでは手品にはなりません。そこで、この節ではハンカチと主役の座が交互に入れ替わるような、そんな手品師的テクニックを披露することにしましょう。

前節でプレーンの分割をやりましたが、今回もテーマは同じプレーンの分割です。ただし、内容も用途もまったく違います。どのように違うか、それは次のプログラムを実行すれば一目瞭然です。マシン語(LIST 3.10)をロードしてから、BASIC(LIST 3.11)を走らせてください。プログラムは全機種対応ですが、PC-8801 mk II以前では速度的に荷が重いことを断わっておきます。

LIST 3.10

```
                        ORG     0B700H

B700 F3                 DI                    ・割込み禁止
B701 DB71               IN      A,(71H)
B703 F5                 PUSH    AF            ・メインROMセレクト
B704 3EFF               LD      A,0FFH
B706 D371               OUT     (71H),A
B708 3E01               LD      A,1           ・テキスト画面オフ
B70A D353               OUT     (53H),A
B70C 3AD779             LD      A,(79D7H)
B70F FE33               CP      '3'
B711 3806               JR      C,PALCC       ・デジタルカラー・モードにする
B713 DB32               IN      A,(32H)
B715 E6DF               AND     11011111B
B717 D332               OUT     (32H),A
B719 2158B7     PALCC:  LD      HL,PALDT
B71C 015408             LD      BC,854H
B71F 7E         PALCL:  LD      A,(HL)
B720 ED79               OUT     (C),A         ・パレットを変更
B722 0C                 INC     C
B723 23                 INC     HL
B724 10F9               DJNZ    PALCL
B726 CD63B7             CALL    MKP3D         ・パターンデータ作成
B729 D35C               OUT     (5CH),A
```

```
B72B CDBCB7              CALL   FLCLS          ・B面に背景を作成
B72E CD96B7              CALL   BACKD
B731 D35F                OUT    (5FH),A
B733 CDDCB7              CALL   DP3GP          ・G面にパターンを作成し表示（パレット変更）
B736 CD4DB8    MLOOP:    CALL   KSCAN          ・移動用キースキャン
B739 CDCAB7              CALL   DP3RP          ・R面にパターンを作成し表示（パレット変更）
B73C CD4DB8              CALL   KSCAN          ・移動用キースキャン
B73F CDDCB7              CALL   DP3GP          ・G面にパターンを作成し表示（パレット変更）
B742 DB09                IN     A,(9)
B744 07                  RLCA                  ・[ESC] が押されていなければMLOOPへ
B745 38EF                JR     C,MLOOP
B747 AF                  XOR    A              ・テキスト画面オン
B748 D353                OUT    (53H),A
B74A 015408              LD     BC,854H
B74D ED79      PAL07:    OUT    (C),A
B74F 0C                  INC    C              ・パレット・イニシャライズ
B750 3C                  INC    A
B751 10FA                DJNZ   PAL07
B753 F1                  POP    AF             ・ROMの状態を元に戻す
B754 D371                OUT    (71H),A
B756 FB                  EI                    ・割込み許可
B757 C9                  RET                   ・BASICへ戻る

B758 00010000  PALDT:    DB     0,1,0,0,0,0,4,4  ・パレット変更用データ
B75C 00000404
B760 00        PAT3D:    DB     0              ・パターン番号
B761 00        XPO3D:    DB     0              ・パターンのX座標
B762 00        YPO3D:    DB     0              ・パターンのY座標

B763 2100C0    MKP3D:    LD     HL,0C000H
B766 1181B8              LD     DE,DAT3D
B769 0608                LD     B,8
B76B CD73B7              CALL   M3DSB
B76E 2180CC              LD     HL,0C000H+80*40
B771 0606                LD     B,6
B773 C5        M3DSB:    PUSH   BC
B774 E5                  PUSH   HL
B775 0E28                LD     C,40
B777 060A      M3DL0:    LD     B,10
B779 D35C      M3DL1:    OUT    (5CH),A
B77B 7E                  LD     A,(HL)
B77C D35F                OUT    (5FH),A
B77E 12                  LD     (DE),A
B77F 23                  INC    HL
B780 13                  INC    DE             ・画面からパターンデータを作成
B781 10F6                DJNZ   M3DL1
B783 3E46                LD     A,80-10
B785 85                  ADD    A,L
B786 3001                JR     NC,M3NIH
B788 24                  INC    H
B789 6F        M3NIH:    LD     L,A
B78A 0D                  DEC    C
B78B 20EA                JR     NZ,M3DL0
B78D E1                  POP    HL
B78E 010A00              LD     BC,10
B791 09                  ADD    HL,BC
B792 C1                  POP    BC
B793 10DE                DJNZ   M3DSB
B795 C9                  RET

B796 0EC8      BACKD:    LD     C,200
B798 2101C0              LD     HL,0C001H
B79B 0627      BAKL0:    LD     B,39
B79D 3601      BAKL1:    LD     (HL),1
B79F 23                  INC    HL
B7A0 23                  INC    HL
B7A1 10FA                DJNZ   BAKL1
B7A3 23                  INC    HL
B7A4 23                  INC    HL
B7A5 0D                  DEC    C
```

```
B7A6 20F3                JR     NZ,BAKL0               ・背景の表示
B7A8 2180C2              LD     HL,0C000H+80*8
B7AB 113002              LD     DE,80*7
B7AE 0E18                LD     C,24
B7B0 0650      BAKL2:    LD     B,80
B7B2 3655      BAKL3:    LD     (HL),01010101B
B7B4 23                  INC    HL
B7B5 10FB                DJNZ   BAKL3
B7B7 19                  ADD    HL,DE
B7B8 0D                  DEC    C
B7B9 20F5                JR     NZ,BAKL2
B7BB C9                  RET

B7BC 2100C0    FLCLS:    LD     HL,0C000H
B7BF 1101C0              LD     DE,0C001H
B7C2 3600                LD     (HL),0                 ・画面クリア・サブルーチン
B7C4 017F3E              LD     BC,80*200-1
B7C7 EDB0                LDIR
B7C9 C9                  RET

B7CA 3E5D      DP3RP:    LD     A,5DH
B7CC CDEEB7              CALL   DIP3D
B7CF AF                  XOR    A
B7D0 D358                OUT    (58H),A
B7D2 3C                  INC    A                      ・R面にパターンを作成し表示（パレット変更）
B7D3 D359                OUT    (59H),A
B7D5 3E04                LD     A,4
B7D7 D356                OUT    (56H),A
B7D9 D357                OUT    (57H),A
B7DB C9                  RET

B7DC 3E5E      DP3GP:    LD     A,5EH
B7DE CDEEB7              CALL   DIP3D
B7E1 AF                  XOR    A
B7E2 D356                OUT    (56H),A
B7E4 3C                  INC    A                      ・G面にパターンを作成し表示（パレット変更）
B7E5 D357                OUT    (57H),A
B7E7 3E04                LD     A,4
B7E9 D358                OUT    (58H),A
B7EB D359                OUT    (59H),A
B7ED C9                  RET

B7EE 322EB8    DIP3D:    LD     (OUTPO+1),A
B7F1 4F                  LD     C,A
B7F2 ED79                OUT    (C),A
B7F4 CDBCB7              CALL   FLCLS
B7F7 D35F                OUT    (5FH),A
B7F9 3A60B7              LD     A,(PAT3D)
B7FC 3C                  INC    A
B7FD FE1B                CP     27
B7FF 3801                JR     C,PATOK
B801 AF                  XOR    A
B802 3260B7    PATOK:    LD     (PAT3D),A
B805 FE0E                CP     14
B807 3803                JR     C,PTIA1
B809 D61B                SUB    27
B80B 2F                  CPL
B80C 3C        PTIA1:    INC    A
B80D 21F1B6              LD     HL,DAT3D-10*40
B810 119001              LD     DE,10*40
B813 19        GPADL:    ADD    HL,DE
B814 3D                  DEC    A
B815 20FC                JR     NZ,GPADL
B817 E5                  PUSH   HL
B818 ED4B61B7            LD     BC,(XPO3D)             ・指定プレーンにパターンを表示
B81C 04                  INC    B
B81D 21C0BE              LD     HL,0C000H-80*4
B820 114001              LD     DE,80*4
B823 19        BCADL:    ADD    HL,DE
B824 10FD                DJNZ   BCADL
```

```
B826 09                 ADD     HL,BC
B827 D1                 POP     DE
B828 0E28               LD      C,40
B82A 060A      DP3L0:   LD      B,10
B82C 1A        DP3L1:   LD      A,(DE)
B82D D35D      OUTPO:   OUT     (5DH),A
B82F 77                 LD      (HL),A
B830 D35F               OUT     (5FH),A
B832 23                 INC     HL
B833 13                 INC     DE
B834 10F6               DJNZ    DP3L1
B836 3E46               LD      A,80-10
B838 85                 ADD     A,L
B839 3001               JR      NC,D3NIH
B83B 24                 INC     H
B83C 6F        D3NIH:   LD      L,A
B83D 0D                 DEC     C
B83E 20EA               JR      NZ,DP3L0
B840 DB40      VWAIT:   IN      A,(40H)
B842 E620               AND     00100000B
B844 20FA               JR      NZ,VWAIT
B846 DB40      VWAT2:   IN      A,(40H)          ・ウェイトを取ってリターン
B848 E620               AND     00100000B
B84A 28FA               JR      Z,VWAT2
B84C C9                 RET

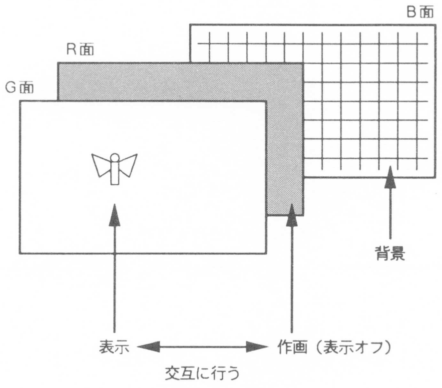
B84D 010000    KSCAN:   LD      BC,0
B850 DB00               IN      A,(0)
B852 CB57               BIT     2,A
B854 2002               JR      NZ,KSCN1
B856 0E01               LD      C,1
B858 CB67      KSCN1:   BIT     4,A
B85A 2002               JR      NZ,KSCN2
B85C 06FF               LD      B,-1
B85E CB77      KSCN2:   BIT     6,A
B860 2002               JR      NZ,KSCN3
B862 0601               LD      B,1
B864 DB01      KSCN3:   IN      A,(1)
B866 0F                 RRCA
B867 3802               JR      C,KSCN4          ・[2] [4] [6] [8] をキースキャンし、パターンの
B869 0EFF               LD      C,-1               座標を変更する
B86B 3A61B7    KSCN4:   LD      A,(XPO3D)
B86E 80                 ADD     A,B
B86F FE47               CP      71
B871 3003               JR      NC,YMCHK
B873 3261B7             LD      (XPO3D),A
B876 3A62B7    YMCHK:   LD      A,(YPO3D)
B879 81                 ADD     A,C
B87A FE29               CP      41
B87C D0                 RET     NC
B87D 3262B7             LD      (YPO3D),A
B880 C9                 RET

B881=          DAT3D:   EQU     $                ・パターンデータエリア
```

LIST 3.11

```
110 CLS 2:X=14:Y=0:B=2:C=0
120 FOR N=0 TO 560 STEP 80
130 GOSUB 210
140 NEXT
150 C=40
160 FOR N=0 TO 400 STEP 80
170 GOSUB 210
180 NEXT
190 CLEAR,&HB5FF:A=&HB700:CALL A
200 END
210 CIRCLE(N+39,C+10),5,1
220 LINE(N+35,C+12)-(N+43,C+30),1,B
230 LINE(N+X+2,C+Y)-(N+35,C+16),1
240 LINE(N+X+2,C+Y)-(N+X,C+Y+13),1
250 LINE(N+X,C+Y+13)-(N+35,C+21),1
260 LINE(N+76-X,C+Y)-(N+43,C+16),1
270 LINE(N+76-X,C+Y)-(N+78-X,C+Y+13),1
280 LINE(N+78-X,C+Y+13)-(N+43,C+21),1
290 Y=Y+2:X=X-B:IF X=0 THEN B=-B
300 RETURN
```

鳥ともマイクのオバケとも取れる妙なものが現れ、テンキーにて自由に動かすことができます。これは単なる線画パターンですが、実際の用途は見ての通り３Ｄ（３次元）の画面処理に使われることが多く、それ以外の用途はあまりありません。画面処理の原理は、図3-3-1に示されているようにＢ面が背景専用、Ｒ面とＧ面は交互に作画（表示オフ）／表示を繰り返すようになっています。つまり、Ｒ面が表示中にはＧ面は作画、Ｇ面が表示中にはＲ面が作画というわけです。本格的な３Ｄ処理では、計算により座標を求めて１本ずつ線を引いていきますから、一般的な消去／表示という方法では画面がチラついてしまうのです。また、このように頻繁にパレット変更をする場合は、「1-4　パレットいろいろ」で述べた理由により垂直同期を取らなければなりません。



パレット番号	G面（メイン部）	R面（メイン部）	B面（背景）		
0	0	0	0	→ 0	背景色
1	0	0	1	→ 1	背景色
2	0	1	0	→ 表示中＝4／作画中＝0	
3	0	1	1	→ 表示中＝4／作画中＝1	
4	1	0	0	→ 作画中＝0／表示中＝4	
5	1	0	1	→ 作画中＝1／表示中＝4	
6	1	1	0	→ 4	常時表示
7	1	1	1	→ 4	常時表示

図3-3-1　３Ｄの画面処理例

今回、旧パターンを消去するのに画面全体をクリアしていますが、これは３Ｄ処理を想定してのものです。というのは、３Ｄ画面はサイズが不定のため、消去は表

示エリア全体をクリアするか、線を1本ずつ消していくかしかないからです。速度とプログラムの効率を考えれば、普通は前者を採用することになります。

なお、画面処理以外の3Dテクニックには触れていませんが、基本は座標を求める計算式を確立し、2点間をラインで結ぶことができればいいのです。とはいえ、それはそれで一冊の本になるべき大きなテーマです。興味のある方は3Dの専門書を読み、その上でこの節の画面処理テクニックを応用してください。ただし、ゲームのデモ画面などにおける3D表示では、計算後の座標だけをデータとして持っておくという方法も有効です。その場合は、高速ラインルーチンだけを作成すればOKです。

さて、このプログラムでは背景が固定されていましたが、実際の3D画面では動く背景が必要なこともあります。当然、それは『スカイ・ブルーザー』のような平面スクロール背景などではなく、メイン画面同様3Dで動くものでなければなりません。しかし、これをB面だけで実現しようとすれば、今度は背景に消去／表示のチラつきが現れることになります。そこで、B面にちょっとした細工をし、R面（またはG面）だけで背景とメイン部を表現できるよう工夫をしたのが次のプログラムです。マシン語（LIST 3.12）をロードしたら、BASICプログラム（LIST 3.13）を走らせてください。

LIST 3.12

```
                          ORG     0B700H

B700 F3                   DI                      ・割込み禁止
B701 DB71                 IN      A,(71H)         ┐
B703 F5                   PUSH    AF              │・メインROMセレクト
B704 3EFF                 LD      A,0FFH          │
B706 D371                 OUT     (71H),A         ┘
B708 3E01                 LD      A,1             ┐・テキスト画面オフ
B70A D353                 OUT     (53H),A         ┘
B70C 3AD779               LD      A,(79D7H)       ┐
B70F FE33                 CP      '3'             │
B711 3806                 JR      C,PALCC         │・デジタルカラー・モードにする
B713 DB32                 IN      A,(32H)         │
B715 E6DF                 AND     11011111B       │
B717 D332                 OUT     (32H),A         ┘
B719 2160B7       PALCC:  LD      HL,PALDT        ┐
B71C 015408               LD      BC,854H         │
B71F 7E           PALCL:  LD      A,(HL)          │
B720 ED79                 OUT     (C),A           │・パレットを変更
B722 0C                   INC     C               │
B723 23                   INC     HL              │
B724 10F9                 DJNZ    PALCL           ┘
B726 CD6DB7               CALL    MKP3D           ・パターンデータ作成
B729 D35C                 OUT     (5CH),A         ┐
B72B 3EAA                 LD      A,10101010B     │
B72D 2100C0               LD      HL,0C000H       │
B730 1101C0               LD      DE,0C001H       │・B面に背景を作成
B733 77                   LD      (HL),A          │
B734 017F3E               LD      BC,80*200-1     │
B737 EDB0                 LDIR                    │
B739 D35F                 OUT     (5FH),A         ┘
B73B CDB5B7               CALL    DP3GP           ・G面にパターンを作成し表示（パレット変更）
B73E CD94B8       MLOOP:  CALL    DIRCK           ・座標移動と方向チェック
B741 CDA0B7               CALL    DP3RP           ・R面にパターンを作成し表示（パレット変更）
B744 CD94B8               CALL    DIRCK           ・座標移動と方向チェック
B747 CDB5B7               CALL    DP3GP           ・G面にパターンを作成し表示（パレット変更）
```

```
B74A DB09               IN     A,(9)           ┐
B74C 07                 RLCA                   │・[ESC] が押されていなければMLOOPへ
B74D 38EF               JR     C,MLOOP         ┘
B74F AF                 XOR    A               ┐
B750 D353               OUT    (53H),A         ┘・テキスト画面オン
B752 015408             LD     BC,854H         ┐
B755 ED79       PAL07:  OUT    (C),A           │
B757 0C                 INC    C               │・パレット・イニシャライズ
B758 3C                 INC    A               │
B759 10FA               DJNZ   PAL07           ┘
B75B F1                 POP    AF              ┐
B75C D371               OUT    (71H),A         ┘・ROMの状態を元に戻す
B75E FB                 EI                      ・割込み許可
B75F C9                 RET                     ・BASICへ戻る

B760 00000000   PALDT:  DB     0,0,0,0,0,0,4,5  ・パレット変更用データ
B764 00000405
B768 00         PAT3D:  DB     0                ・パターン番号
B769 00         DIR3D:  DB     0                ・パターンの方向
B76A 14         YPO3D:  DB     20               ・パターンのY座標
B76B 02         BAKLL:  DB     00000010B        ・背景の左側データ
B76C 80         BAKRR:  DB     10000000B        ・背景の右側データ

B76D 2100C0     MKP3D:  LD     HL,0C000H       ┐
B770 11B5B8             LD     DE,DAT3D        │
B773 0608               LD     B,8             │
B775 CD7DB7             CALL   M3DSB           │
B778 2180CC             LD     HL,0C000H+80*40 │
B77B 0606               LD     B,6             │
B77D C5         M3DSB:  PUSH   BC              │
B77E E5                 PUSH   HL              │
B77F 0E28               LD     C,40            │
B781 060A       M3DL0:  LD     B,10            │
B783 D35C       M3DL1:  OUT    (5CH),A         │
B785 7E                 LD     A,(HL)          │
B786 D35F               OUT    (5FH),A         │
B788 12                 LD     (DE),A          │
B789 23                 INC    HL              │
B78A 13                 INC    DE              │・画面からパターンデータを作成
B78B 10F6               DJNZ   M3DL1           │
B78D 3E46               LD     A,80-10         │
B78F 85                 ADD    A,L             │
B790 3001               JR     NC,M3NIH        │
B792 24                 INC    H               │
B793 6F         M3NIH:  LD     L,A             │
B794 0D                 DEC    C               │
B795 20EA               JR     NZ,M3DL0        │
B797 E1                 POP    HL              │
B798 010A00             LD     BC,10           │
B79B 09                 ADD    HL,BC           │
B79C C1                 POP    BC              │
B79D 10DE               DJNZ   M3DSB           │
B79F C9                 RET                    ┘

B7A0 3E5D       DP3RP:  LD     A,5DH           ┐
B7A2 CDCAB7             CALL   DIP3D           │
B7A5 CD87B8             CALL   VWAIT           │
B7A8 AF                 XOR    A               │
B7A9 D358               OUT    (58H),A         │
B7AB D359               OUT    (59H),A         │・背景の表示
B7AD 3E04               LD     A,4             │
B7AF D356               OUT    (56H),A         │
B7B1 3C                 INC    A               │
B7B2 D357               OUT    (57H),A         │
B7B4 C9                 RET                    ┘

B7B5 3E5E       DP3GP:  LD     A,5EH           ┐
B7B7 CDCAB7             CALL   DIP3D           │
B7BA CD87B8             CALL   VWAIT           │
B7BD AF                 XOR    A               │
```

```
B7BE D356               OUT     (56H),A
B7C0 D357               OUT     (57H),A            ・画面クリア・サブルーチン
B7C2 3E04               LD      A,4
B7C4 D358               OUT     (58H),A
B7C6 3C                 INC     A
B7C7 D359               OUT     (59H),A
B7C9 C9                 RET

B7CA 3220B8   DIP3D:    LD      (OUTPO+1),A
B7CD 4F                 LD      C,A
B7CE ED79               OUT     (C),A
B7D0 CD32B8             CALL    BAKMM
B7D3 D35F               OUT     (5FH),A
B7D5 3A68B7             LD      A,(PAT3D)
B7D8 3C                 INC     A
B7D9 FE1B               CP      27
B7DB 3801               JR      C,PATOK
B7DD AF                 XOR     A
B7DE 3268B7   PATOK:    LD      (PAT3D),A
B7E1 FE0E               CP      14
B7E3 3803               JR      C,PTIA1
B7E5 D61B               SUB     27
B7E7 2F                 CPL
B7E8 3C       PTIA1:    INC     A
B7E9 2125B7             LD      HL,DAT3D-10*40
B7EC 119001             LD      DE,10*40
B7EF 19       GPADL:    ADD     HL,DE
B7F0 3D                 DEC     A
B7F1 20FC               JR      NZ,GPADL
B7F3 E5                 PUSH    HL
B7F4 1123C0             LD      DE,0C023H
B7F7 214600             LD      HL,80-10
B7FA 3A69B7             LD      A,(DIR3D)
B7FD B7                 OR      A
B7FE 2806               JR      Z,DIR30
B800 11A3CC             LD      DE,0C023H+80*40
B803 21A6FF             LD      HL,-80-10
B806 222AB8   DIR30:    LD      (DIPON+1),HL       •R面（またはG面）に背景／パターンを作成し
B809 3A6AB7             LD      A,(YPO3D)          表示（パレット変更）
B80C 6F                 LD      L,A
B80D 2600               LD      H,0
B80F 29                 ADD     HL,HL
B810 29                 ADD     HL,HL
B811 29                 ADD     HL,HL
B812 29                 ADD     HL,HL
B813 44                 LD      B,H
B814 4D                 LD      C,L
B815 29                 ADD     HL,HL
B816 29                 ADD     HL,HL
B817 09                 ADD     HL,BC
B818 19                 ADD     HL,DE
B819 D1                 POP     DE
B81A 01280A             LD      BC,10*100H+40
B81D C5       DP3L0:    PUSH    BC
B81E 1A       DP3L1:    LD      A,(DE)
B81F D35D     OUTPO:    OUT     (5DH),A
B821 B6                 OR      (HL)
B822 77                 LD      (HL),A
B823 D35F               OUT     (5FH),A
B825 23                 INC     HL
B826 13                 INC     DE
B827 10F5               DJNZ    DP3L1
B829 014600   DIPON:    LD      BC,80-10
B82C 09                 ADD     HL,BC
B82D C1                 POP     BC
B82E 0D                 DEC     C
B82F 20EC               JR      NZ,DP3L0
B831 C9                 RET

B832 216BB7   BAKMM:    LD      HL,BAKLL
```

```
B835 3A69B7          LD     A,(DIR3D)
B838 B7              OR     A
B839 200C            JR     NZ,ROTA1
B83B CB06            RLC    (HL)
B83D CB06            RLC    (HL)
B83F 56              LD     D,(HL)
B840 23              INC    HL
B841 CB0E            RRC    (HL)
B843 CB0E            RRC    (HL)
B845 180A            JR     BAKM1
B847 CB0E    ROTA1:  RRC    (HL)
B849 CB0E            RRC    (HL)
B84B 56              LD     D,(HL)
B84C 23              INC    HL
B84D CB06            RLC    (HL)
B84F CB06            RLC    (HL)
B851 5E      BAKM1:  LD     E,(HL)
B852 210AC0          LD     HL,0C000H+10
B855 0EC8            LD     C,200
B857 7A      MBALP:  LD     A,D
B858 F6A0            OR     10100000B
B85A 77              LD     (HL),A
B85B 23              INC    HL
B85C 061B            LD     B,27
B85E 72      MBAL0:  LD     (HL),D
B85F 23              INC    HL
B860 10FC            DJNZ   MBAL0          ・背景をローテートして指定プレーンに表示
B862 7A              LD     A,D
B863 F60A            OR     00001010B
B865 77              LD     (HL),A
B866 23              INC    HL
B867 3600            LD     (HL),0
B869 23              INC    HL
B86A 3600            LD     (HL),0
B86C 23              INC    HL
B86D 7B              LD     A,E
B86E F6A0            OR     10100000B
B870 77              LD     (HL),A
B871 23              INC    HL
B872 061B    RBVAL:  LD     B,27
B874 73      MBAL1:  LD     (HL),E
B875 23              INC    HL
B876 10FC            DJNZ   MBAL1
B878 7B              LD     A,E
B879 F60A            OR     00001010B
B87B 77              LD     (HL),A
B87C 3E15            LD     A,21
B87E 85              ADD    A,L
B87F 3001            JR     NC,MBNIH
B881 24              INC    H
B882 6F      MBNIH:  LD     L,A
B883 0D              DEC    C
B884 20D1            JR     NZ,MBALP
B886 C9              RET

B887 DB40    VWAIT:  IN     A,(40H)
B889 E620            AND    00100000B
B88B 20FA            JR     NZ,VWAIT
B88D DB40    VWAT2:  IN     A,(40H)        ・ウェイトルーチン
B88F E620            AND    00100000B
B891 28FA            JR     Z,VWAT2
B893 C9              RET

B894 216AB7  DIRCK:  LD     HL,YPO3D
B897 3A69B7          LD     A,(DIR3D)
B89A B7              OR     A
B89B 2808            JR     Z,DIR00
B89D 7E              LD     A,(HL)
B89E 3C              INC    A
B89F FEA1            CP     161
```

```
B8A1 2809               JR      Z,DIRRV
B8A3 77                 LD      (HL),A
B8A4 C9                 RET                       ・パターン座標を変更し移動方向をチェック
B8A5 7E         DIR00:  LD      A,(HL)
B8A6 D601               SUB     1
B8A8 3802               JR      C,DIRRV
B8AA 77                 LD      (HL),A
B8AB C9                 RET
B8AC 3A69B7     DIRRV:  LD      A,(DIR3D)
B8AF EE01               XOR     1
B8B1 3269B7             LD      (DIR3D),A
B8B4 C9                 RET

B8B5=           DAT3D:  EQU     $                 ・パターンデータエリア
```

LIST 3.13

```
110 CLS 2:X=14:Y=0:B=2:C=0
120 FOR N=0 TO 560 STEP 80
130 M=N:GOSUB 210:M=N+1:GOSUB 210
135 Y=Y+2:X=X-B:IF X=0 THEN B=-B
140 NEXT
150 C=40
160 FOR N=0 TO 400 STEP 80
170 M=N:GOSUB 210:M=N+1:GOSUB 210
175 Y=Y+2:X=X-B:IF X=0 THEN B=-B
180 NEXT
190 CLEAR,&HB5FF:A=&HB700:CALL A
200 END
210 CIRCLE(M+39,C+10),5,1
220 LINE(M+35,C+12)-(M+43,C+30),1,B
230 LINE(M+X+2,C+Y)-(M+35,C+16),1
240 LINE(M+X+2,C+Y)-(M+X,C+Y+13),1
250 LINE(M+X,C+Y+13)-(M+35,C+21),1
260 LINE(M+76-X,C+Y)-(M+43,C+16),1
270 LINE(M+76-X,C+Y)-(M+78-X,C+Y+13),1
280 LINE(M+78-X,C+Y+13)-(M+43,C+21),1
300 RETURN
```

別々の色で背景とメイン部が表示されていますが、これらはどちらも同じプレーンに描かれたものです。よく見ると、背景はカラー5/0のタイリング、メイン部はカラー5/4のタイリングで描かれています。実は、このタイリングカラーを実現している秘密がB面にあるのです。では、ESCキーでプログラムの実行を停止してください。

パレットが初期化され、B面はすべて10101010Bで埋め尽くされていることがわかります。これがB面の秘密です。こうしておくと、R面（またはG面）にデータをストアすればカラー5/4のタイリングとすることができるし、データを10101010BとANDを取ってストアすれば、カラー5/0のタイリングとすることができます（図3-3-2参照）。

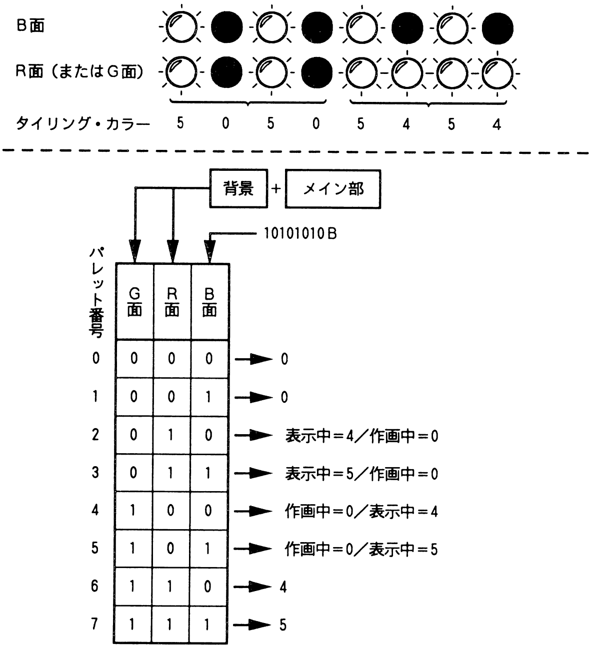
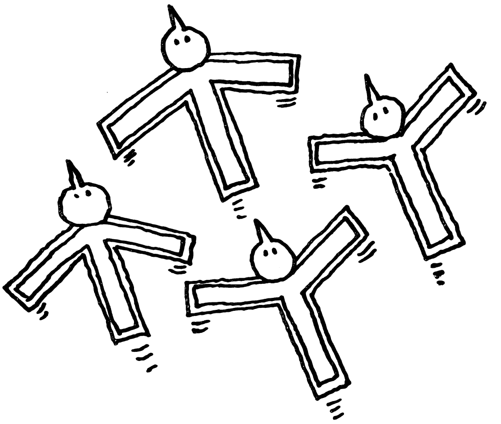


図 3-3-2　3D用タイリング・カラー

ここではテスト的な背景ですから、特に背景表示の際にANDを取るという作業もありませんが、実際には背景はすべて10101010BとANDを取り、メイン部をORで重ね合わせるという手順となります。その際、メイン部のラインは必ず横2ドット以上で構成しなければならないということに注意してください。こうしないと、条件によっては背景と同化してしまうことがあるからです。

さて、このプログラムにはテンキー操作を省いた代わりに、パターンの上下反転表示を取り入れてあります。上から下へ表示するか、下から上へ表示するか、たったそれだけの違いですが、データ活用のミニ・テクニックとして覚えておくと便利です。



3-4 おしゃれな表示/消去

出会いと別れ、入学と卒業、生と死……。何事において、始まりがあるものには終わりがあります。そして、ヒーロー（あるいはヒロイン）と呼ばれる人たちは、それが実にカッコイイのです。ましてや映画やテレビなどでは、現実と違いドジや失敗があってもやり直しがききますから、われわれがマネをしようと思ってもそうウマクいくものではありません。しかし、現実はダメでもやり直しのきくプログラムの世界でなら、いくらでもカッコよさを追求できるではありませんか。

画面に表示されたものは、必ず消え去る運命にあります。ゲームも一種のエンターテインメントの世界ですから、ここにカッコよさを取り入れるのは制作者の務めです。もしかすると、それだけでゲームが見違えるようになるかもしれません。この節では、そんな表示／消去の化粧品をいくつか集めてみました。

さて、一口に表示／消去といっても、基本的にデータの不要な消去に対し、表示にはグラフィック・データが必要です。しかも、それはゲームの顔であるタイトル画面、アドベンチャー・ゲームなどのシーン画面、パターンによるマップ画面……等、状況に応じてデータ構造が違います。例えば、タイトルやエンディング画面では、表示過程を特に効果的に見せるために大量のデータをベタで持つこともありますが、アドベンチャー・ゲームのシーン画面ではデータに圧縮をかけるのが普通です。また、パターンによるマップ画面ではパターンのサイズやマップデータの構造も問題となるでしょう。

グラフィック・データの圧縮／展開については、テーマとして大き過ぎるだけでなく展開が圧縮の逆の作業であることを考えると化粧にも限界があります。そのため、「**3-1　テキスト画面マスク**」で取り上げたテキスト画面によるブラインド処理が最適の化粧品となっています。となると残るは2つですが、この2つのグラフィック・データはサイズが違うだけで基本的には同じ構造（B／R／G順に並んだもの）です。そこで、具体的な化粧に入る前に、まずパターンによるマップ画面とはどのようなものなのか、マップデータの構造から確認することにしましょう。

これは次の章で解説するスクロール・テクニックにも関連していることなのですが、いわゆるアクションRPGで用いられている画面は、横16ドット×縦8ドットのマップパターンによって構成されているのが一般的です。逆重ね合わせ（部分的にキャラクタより背景が上に表示される）などの高度なテクニックを使う場合は、マップパターンのデータにもくり抜き用の透明データが必要ですが、ここではそれは考えないことにします。この小さなマップパターンの組み合わせがマップ画面となり、それが連続することによってスクロール・マップデータとなるわけです（図3-4-1）。

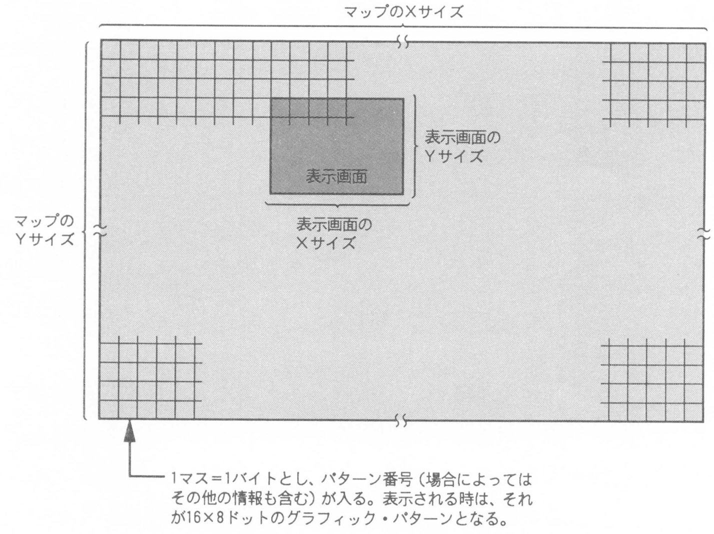


図3-4-1　マップデータの構造

スクロールの手法はともかく、スタートはマップデータの一部を画面に表示することです。これは何もゲームのスタート時だけではなく、町やフィールド、ダンジョンとの出入りなど、画面が切り替わるたびに必要になることです。最もシンプルなのは、画面上部から順に表示していく方法ですが、これは表示速度が速くプログラムも簡単ということ以外に魅力はありません。もちろん、このことも画面表示の重要なファクターですが、今回はここにカッコよさを求めているわけですから、それ以上に「見せる」という工夫が要求されています。

そこで、内容はあくまでもテスト的なものですが、サイズフリーのマップとサイズフリーの表示画面に対応したものを2組、そしてタイトルなどベタ・データに対応したものを1組、計3組の化粧プログラムを用意してみました。では、プログラム（LIST 3.14）を実行してください。

```
LINE(0,0)-(639,199),7,B
LINE(125,30)-(514,169),7,B
PAINT(10,10),5,7
CLEAR,&HB5FF:A=&HB700:CALL A
```

LIST 3.14

```
                          ORG       0B700H

B700 F3                   DI                          ・割込み禁止
B701 ED735AB7             LD        (SPPOP+1),SP    ┐
B705 3100B7               LD        SP,0B700H       ┘・SPの退避と設定
B708 AF                   XOR       A               ┐
B709 322EB7               LD        (DCEND+1),A     ┘・消去／表示フラグをイニシャライズ
B70C 3C                   INC       A               ┐
B70D D353                 OUT       (53H),A         ┘・テキスト画面オフ
```

```
B70F CD84B7              CALL   SDISP             ・初期画面表示
B712 3AC2E6              LD     A,(0E6C2H)       ┐
B715 F602                OR     2                ├・裏RAMセレクト
B717 D331                OUT    (31H),A          ┘
B719 110010              LD     DE,1000H         ┐
B71C D35C                OUT    (5CH),A          │
B71E CD5EB7              CALL   MDATA            │
B721 D35D                OUT    (5DH),A          ├・1000H番地からに初期画面をベタでデータ化
B723 CD5EB7              CALL   MDATA            │
B726 D35E                OUT    (5EH),A          │
B728 CD5EB7              CALL   MDATA            │
B72B D35F                OUT    (5FH),A          ┘
B72D 3E00       DCEND:   LD     A,0              ┐
B72F EE01                XOR    1                ├・消去／表示フラグを反転
B731 322EB7              LD     (DCEND+1),A      ┘
B734 DB00       DCEDL:   IN     A,(0)            ┐
B736 3C                  INC    A                ├・[0] [1] [2] が押されなくなるまでループ
B737 20FB                JR     NZ,DCEDL         ┘
B739 212DB7     MLOOP:   LD     HL,DCEND         ┐
B73C E5                  PUSH   HL               ┘・リターンでの戻り先
B73D DB00                IN     A,(0)            ┐
B73F 0F                  RRCA                    ├・[0] が押されていればDISP0へ
B740 D2C2B7              JP     NC,DISP0         ┘
B743 0F                  RRCA                    ┐
B744 D25DB8              JP     NC,DISP1         ┘・[1] が押されていればDISP1へ
B747 0F                  RRCA                    ┐
B748 D249B9              JP     NC,DISP2         ┘・[2] が押されていればDISP2へ
B74B E1                  POP    HL                ・ダミー
B74C DB09                IN     A,(9)            ┐
B74E 07                  RLCA                    ├・[ESC] が押されていなければMLOOPへ
B74F 38E8                JR     C,MLOOP          ┘
B751 3AC2E6              LD     A,(0E6C2H)       ┐
B754 D331                OUT    (31H),A          ┘・メインROMセレクト
B756 AF                  XOR    A                ┐
B757 D353                OUT    (53H),A          ┘・テキスト画面オン
B759 310000     SPPOP:   LD     SP,0              ・SPを元に戻す
B75C FB                  EI                       ・割込み許可
B75D C9                  RET                      ・BASICへ戻る

B75E 2110CA     MDATA:   LD     HL,DVRAM         ┐
B761 3E88                LD     A,GYMAX*8        │
B763 013000     MDATL:   LD     BC,GXMAX*2       │
B766 EDB0                LDIR                    │
B768 012000              LD     BC,80-GXMAX*2    ├・プレーン別に表示画面のデータを転送
B76B 09                  ADD    HL,BC            │
B76C 3D                  DEC    A                │
B76D 20F4                JR     NZ,MDATL         │
B76F C9                  RET                     ┘
B770 DB09       VWCHK:   IN     A,(9)            ┐
B772 07                  RLCA                    │
B773 07                  RLCA                    │
B774 D8                  RET    C                │
B775 DB40       VWAIT:   IN     A,(40H)          │
B777 E620                AND    00100000B        ├・[SPACE] が押されていればウェイトを取る
B779 20FA                JR     NZ,VWAIT         │
B77B DB40       VWAT2:   IN     A,(40H)          │
B77D E620                AND    00100000B        │
B77F 28FA                JR     Z,VWAT2          │
B781 C9                  RET                     ┘

CA10=           DVRAM:   EQU    0CA10H            ・表示アドレス
0018=           GXMAX:   EQU    24                ・表示パターン数 (X)
0011=           GYMAX:   EQU    17                ・表示パターン数 (Y)
1980=           G1VAL:   EQU    GXMAX*2*GYMAX*8   ・表示画面ベタデータ数 (プレーン別)

0018=           MXMAX:   EQU    24                ・マップサイズ (X)
0011=           MYMAX:   EQU    17                ・マップサイズ (Y)
B782 00         MTOPX:   DB     0                 ・表示マップトップ (X)
B783 00         MTOPY:   DB     0                 ・表示マップトップ (Y)
```

```
B784 1110CA    SDISP:  LD      DE,DVRAM
B787 D9                EXX
B788 010000            LD      BC,0
B78B CD0BBA            CALL    MAPAD
B78E 110000            LD      DE,MXMAX-GXMAX
B791 D9                EXX
B792 0E11              LD      C,GYMAX
B794 0618      SDIL0:  LD      B,GXMAX
B796 C5        SDIL1:  PUSH    BC
B797 D5                PUSH    DE
B798 D9                EXX
B799 7E                LD      A,(HL)
B79A 23                INC     HL
B79B D9                EXX
B79C CD23BA            CALL    PTADR
B79F D1                POP     DE
B7A0 D35C              OUT     (5CH),A
B7A2 CD3AB8            CALL    BCDIP          ・マップにしたがって画面に表示する（ノーマ
B7A5 D35D              OUT     (5DH),A          ル）
B7A7 CD3AB8            CALL    BCDIP
B7AA D35E              OUT     (5EH),A
B7AC CD3AB8            CALL    BCDIP
B7AF D35F              OUT     (5FH),A
B7B1 13                INC     DE
B7B2 13                INC     DE
B7B3 C1                POP     BC
B7B4 10E0              DJNZ    SDIL1
B7B6 215002            LD      HL,80*8-GXMAX*2
B7B9 19                ADD     HL,DE
B7BA EB                EX      DE,HL
B7BB D9                EXX
B7BC 19                ADD     HL,DE
B7BD D9                EXX
B7BE 0D                DEC     C
B7BF 20D3              JR      NZ,SDIL0
B7C1 C9                RET

B7C2 010000    DISP0:  LD      BC,0
B7C5 1610              LD      D,GYMAX-1
B7C7 1E18              LD      E,GXMAX
B7C9 7B        DIP0L:  LD      A,E
B7CA B7                OR      A
B7CB C8                RET     Z
B7CC D5                PUSH    DE
B7CD CD08B8    YD0UL:  CALL    BCDP0          ・画面の上側を表示する
B7D0 0C                INC     C
B7D1 1D                DEC     E
B7D2 20F9              JR      NZ,YD0UL
B7D4 0D                DEC     C
B7D5 04                INC     B
B7D6 D1                POP     DE
B7D7 1D                DEC     E

B7D8 7A                LD      A,D
B7D9 B7                OR      A
B7DA C8                RET     Z
B7DB D5                PUSH    DE
B7DC CD08B8    YD0RL:  CALL    BCDP0
B7DF 04                INC     B
B7E0 15                DEC     D              ・画面の右側を表示する
B7E1 20F9              JR      NZ,YD0RL
B7E3 05                DEC     B
B7E4 0D                DEC     C
B7E5 D1                POP     DE
B7E6 15                DEC     D

B7E7 7B                LD      A,E
B7E8 B7                OR      A
B7E9 C8                RET     Z
B7EA D5                PUSH    DE
```

```
B7EB CD08B8    YD0DL:  CALL   BCDP0
B7EE 0D                DEC    C
B7EF 1D                DEC    E              ・画面の下側を表示する
B7F0 20F9              JR     NZ,YD0DL
B7F2 0C                INC    C
B7F3 05                DEC    B
B7F4 D1                POP    DE
B7F5 1D                DEC    E

B7F6 7A                LD     A,D
B7F7 B7                OR     A
B7F8 C8                RET    Z
B7F9 D5                PUSH   DE
B7FA CD08B8    YD0LL:  CALL   BCDP0
B7FD 05                DEC    B
B7FE 15                DEC    D              ・画面の左側を表示する
B7FF 20F9              JR     NZ,YD0LL
B801 04                INC    B
B802 0C                INC    C
B803 D1                POP    DE
B804 15                DEC    D
B805 C3C9B7            JP     DIP0L          ・DIPOLへ

B808 C5        BCDP0:  PUSH   BC
B809 D5                PUSH   DE
B80A CD0BBA            CALL   MAPAD
B80D 7E                LD     A,(HL)
B80E CD23BA            CALL   PTADR
B811 E5                PUSH   HL
B812 CD32BA            CALL   BCADR
B815 E1                POP    HL
B816 D35C              OUT    (5CH),A
B818 CD34B8            CALL   BC0SB
B81B D35D              OUT    (5DH),A
B81D CD34B8            CALL   BC0SB          ・(C, B) の位置にマップを表示（消去）する
B820 D35E              OUT    (5EH),A
B822 CD34B8            CALL   BC0SB
B825 D35F              OUT    (5FH),A
B827 0E06              LD     C,6
B829 10FE      BWLOP:  DJNZ   $
B82B 0D                DEC    C
B82C 20FB              JR     NZ,BWLOP
B82E CD70B7            CALL   VWCHK
B831 D1                POP    DE
B832 C1                POP    BC
B833 C9                RET

B834 3A2EB7    BC0SB:  LD     A,(DCEND+1)
B837 B7                OR     A              ・消去／表示のフラグチェック
B838 2013              JR     NZ,B0CLS
B83A D5        BCDIP:  PUSH   DE
B83B 01FF08            LD     BC,8FFH
B83E EDA0      BC0SL:  LDI
B840 EDA0              LDI
B842 3E4E              LD     A,80-2
B844 83                ADD    A,E
B845 3001              JR     NC,BC0S1       ・マップ表示のサブルーチン
B847 14                INC    D
B848 5F        BC0S1:  LD     E,A
B849 10F3              DJNZ   BC0SL
B84B D1                POP    DE
B84C C9                RET

B84D D5        B0CLS:  PUSH   DE
B84E EB                EX     DE,HL
B84F 114F00            LD     DE,80-1
B852 AF                XOR    A
B853 0608              LD     B,8
B855 77        B0CLP:  LD     (HL),A
B856 23                INC    HL             ・画面上のマップ消去サブルーチン
```

```
B857 77                 LD      (HL),A
B858 19                 ADD     HL,DE
B859 10FA               DJNZ    B0CLP
B85B D1                 POP     DE
B85C C9                 RET

B85D 3A2EB7   DISP1:    LD      A,(DCEND+1)          ・消去／表示のフラグチェック
B860 B7                 OR      A
B861 C2FCB8             JP      NZ,D1CLS
B864 010000             LD      BC,0
B867 CD0BBA             CALL    MAPAD
B86A 110000             LD      DE,MXMAX-GXMAX
B86D 0610               LD      B,2*8
B86F DD21CCB8           LD      IX,V1TOP
B873 E5       D1LOP:    PUSH    HL                   ・画面上にマップをジワジワと表示
B874 D9                 EXX
B875 CD80B8             CALL    D1SUB
B878 CD75B7             CALL    VWAIT
B87B D9                 EXX
B87C E1                 POP     HL
B87D 10F4               DJNZ    D1LOP
B87F C9                 RET

B880 DD7E00   D1SUB:    LD      A,(IX+0)
B883 32A1B8             LD      (D1PON+2),A
B886 DD6E01             LD      L,(IX+1)
B889 DD6602             LD      H,(IX+2)
B88C DD23               INC     IX
B88E DD23               INC     IX
B890 DD23               INC     IX
B892 0E11               LD      C,GYMAX
B894 0618     D1LP0:    LD      B,GXMAX
B896 D9       D1LP1:    EXX
B897 7E                 LD      A,(HL)
B898 23                 INC     HL
B899 D9                 EXX
B89A E5                 PUSH    HL
B89B CD23BA             CALL    PTADR
B89E EB                 EX      DE,HL
B89F FD210000 D1PON:    LD      IY,0
B8A3 FD19               ADD     IY,DE
B8A5 E1                 POP     HL
B8A6 D35C               OUT     (5CH),A              ・1バイトを一定のアルゴリズムで表示
B8A8 FD7E00             LD      A,(IY+0)               IX＝表示する1バイトの位置情報
B8AB 77                 LD      (HL),A                 C＝表示パターン数（Y）
B8AC D35D               OUT     (5DH),A                B＝表示パターン数（X）
B8AE FD7E10             LD      A,(IY+16)              裏HL＝マップアドレス
B8B1 77                 LD      (HL),A                 裏DE＝次マップへの加算量
B8B2 D35E               OUT     (5EH),A                IY＝パターンデータアドレス
B8B4 FD7E20             LD      A,(IY+32)              HL＝表示アドレス
B8B7 77                 LD      (HL),A
B8B8 D35F               OUT     (5FH),A
B8BA 23                 INC     HL
B8BB 23                 INC     HL
B8BC 10D8               DJNZ    D1LP1
B8BE 115002             LD      DE,80*8-GXMAX*2
B8C1 19                 ADD     HL,DE
B8C2 D9                 EXX
B8C3 19                 ADD     HL,DE
B8C4 D9                 EXX
B8C5 CD70B7             CALL    VWCHK
B8C8 0D                 DEC     C
B8C9 20C9               JR      NZ,D1LP0
B8CB C9                 RET

B8CC 00       V1TOP:    DB      0
B8CD 10CA               DW      DVRAM
B8CF 05                 DB      5
B8D0 B1CA               DW      DVRAM+80*2+1
```

```
B8D2 0A                   DB      10
B8D3 A0CB                 DW      DVRAM+80*5
B8D5 0F                   DB      15
B8D6 41CC                 DW      DVRAM+80*7+1
B8D8 04                   DB      4
B8D9 B0CA                 DW      DVRAM+80*2
B8DB 09                   DB      9
B8DC 51CB                 DW      DVRAM+80*4+1
B8DE 0E                   DB      14
B8DF 40CC                 DW      DVRAM+80*7
B8E1 03                   DB      3
B8E2 61CA                 DW      DVRAM+81          ・1バイトの表示位置情報
B8E4 08                   DB      8
B8E5 50CB                 DW      DVRAM+80*4
B8E7 0D                   DB      13
B8E8 F1CB                 DW      DVRAM+80*6+1
B8EA 02                   DB      2
B8EB 60CA                 DW      DVRAM+80
B8ED 07                   DB      7
B8EE 01CB                 DW      DVRAM+80*3+1
B8F0 0C                   DB      12
B8F1 F0CB                 DW      DVRAM+80*6
B8F3 01                   DB      1
B8F4 11CA                 DW      DVRAM+1
B8F6 06                   DB      6
B8F7 00CB                 DW      DVRAM+80*3
B8F9 0B                   DB      11
B8FA A1CB                 DW      DVRAM+80*5+1

B8FC 117000     D1CLS:    LD      DE,80*2-GXMAX*2
B8FF 0E77                 LD      C,01110111B
B901 D9                   EXX
B902 117000               LD      DE,80*2-GXMAX*2
B905 0EEE                 LD      C,11101110B
B907 D9                   EXX
B908 CD11B9               CALL    C1SUB
B90B CD11B9               CALL    C1SUB
B90E CD11B9               CALL    C1SUB
B911 117000     C1SUB:    LD      DE,80*2-GXMAX*2   ・表示画面をジワジワと消去
B914 2110CA               LD      HL,DVRAM
B917 CD27B9               CALL    C1SB2
B91A CB09                 RRC     C
B91C D9                   EXX
B91D 2160CA               LD      HL,DVRAM+80
B920 CD27B9               CALL    C1SB2
B923 CB01                 RLC     C
B925 D9                   EXX
B926 C9                   RET

B927 3E44       C1SB2:    LD      A,GYMAX*4
B929 08         C1SL0:    EX      AF,AF'
B92A 0630                 LD      B,GXMAX*2
B92C D35C       C1SL1:    OUT     (5CH),A
B92E 79                   LD      A,C
B92F A6                   AND     (HL)
B930 77                   LD      (HL),A
B931 D35D                 OUT     (5DH),A
B933 79                   LD      A,C
B934 A6                   AND     (HL)
B935 77                   LD      (HL),A
B936 D35E                 OUT     (5EH),A           ・ドット単位で表示画面を消去する
B938 79                   LD      A,C
B939 A6                   AND     (HL)
B93A 77                   LD      (HL),A
B93B 23                   INC     HL
B93C 10EE                 DJNZ    C1SL1
B93E CD70B7               CALL    VWCHK
B941 19                   ADD     HL,DE
B942 08                   EX      AF,AF'
B943 3D                   DEC     A
```

```
B944 20E3                JR      NZ,C1SL0
B946 D35F                OUT     (5FH),A
B948 C9                  RET

B949 3A2EB7     DISP2:   LD      A,(DCEND+1)
B94C B7                  OR      A                      ・消去／表示のフラグチェック
B94D C2A4B9              JP      NZ,D2CLS
B950 D35C                OUT     (5CH),A
B952 21D00F              LD      HL,1000H-GXMAX*2
B955 CD6BB9              CALL    D2SUB
B958 D35D                OUT     (5DH),A
B95A 215029              LD      HL,1000H+G1VAL-GXMAX*2
B95D CD6BB9              CALL    D2SUB                  ・画面上にマップを斜め表示
B960 D35E                OUT     (5EH),A
B962 21D042              LD      HL,1000H+G1VAL*2-GXMAX*2
B965 CD6BB9              CALL    D2SUB
B968 D35F                OUT     (5FH),A
B96A C9                  RET

B96B 22E8B9     D2SUB:   LD      (GHPON+1),HL
B96E 010000              LD      BC,0
B971 CDE5B9              CALL    GETHD
B974 118019              LD      DE,G1VAL
B977 D5         D2SLP:   PUSH    DE
B978 DD7E00              LD      A,(IX+0)
B97B 77                  LD      (HL),A
B97C 113100              LD      DE,GXMAX*2+1
B97F DD19                ADD     IX,DE
B981 115100              LD      DE,80+1
B984 19                  ADD     HL,DE
B985 0C                  INC     C
B986 79                  LD      A,C
B987 FE30                CP      GXMAX*2
B989 2004                JR      NZ,D2COK               ・プレーン別に斜め表示を行う
B98B 0E00                LD      C,0
B98D 1808                JR      CGTHD
B98F 04         D2COK:   INC     B
B990 78                  LD      A,B
B991 FE88                CP      GYMAX*8
B993 2008                JR      NZ,D2BOK
B995 0600                LD      B,0
B997 CDE5B9     CGTHD:   CALL    GETHD
B99A CD70B7              CALL    VWCHK
B99D D1         D2BOK:   POP     DE
B99E 1B                  DEC     DE
B99F 7B                  LD      A,E
B9A0 B2                  OR      D
B9A1 20D4                JR      NZ,D2SLP
B9A3 C9                  RET

B9A4 D35E       D2CLS:   OUT     (5EH),A
B9A6 CDB6B9              CALL    C2SUB
B9A9 D35D                OUT     (5DH),A
B9AB CDB6B9              CALL    C2SUB                  ・表示画面を斜め消去
B9AE D35C                OUT     (5CH),A
B9B0 CDB6B9              CALL    C2SUB
B9B3 D35F                OUT     (5FH),A
B9B5 C9                  RET

B9B6 010000     C2SUB:   LD      BC,0
B9B9 CDF6B9              CALL    V2ADR
B9BC 118019              LD      DE,G1VAL
B9BF D5         C2SLP:   PUSH    DE
B9C0 3600                LD      (HL),0
B9C2 115100              LD      DE,80+1
B9C5 19                  ADD     HL,DE
B9C6 0C                  INC     C
B9C7 79                  LD      A,C
B9C8 FE30                CP      GXMAX*2
B9CA 2004                JR      NZ,C2COK
B9CC 0E00                LD      C,0
```

```
B9CE 1808                JR     CV2AD       ・プレーン別に斜め消去を行う
B9D0 04         C2COK:   INC    B
B9D1 78                  LD     A,B
B9D2 FE88                CP     GYMAX*8
B9D4 2008                JR     NZ,C2BOK
B9D6 0600                LD     B,0
B9D8 CDF6B9     CV2AD:   CALL   V2ADR
B9DB CD70B7              CALL   VWCHK
B9DE D1         C2BOK:   POP    DE
B9DF 1B                  DEC    DE
B9E0 7B                  LD     A,E
B9E1 B2                  OR     D
B9E2 20DB                JR     NZ,C2SLP
B9E4 C9                  RET

B9E5 78         GETHD:   LD     A,B
B9E6 3C                  INC    A
B9E7 21D00F     GHPON:   LD     HL,1000H-GXMAX*2
B9EA 113000              LD     DE,GXMAX*2
B9ED 19         GTHLP:   ADD    HL,DE
B9EE 3D                  DEC    A
B9EF 20FC                JR     NZ,GTHLP
B9F1 59                  LD     E,C
B9F2 19                  ADD    HL,DE
B9F3 E5                  PUSH   HL
B9F4 DDE1                POP    IX
B9F6 68         V2ADR:   LD     L,B
B9F7 2600                LD     H,0
B9F9 29                  ADD    HL,HL
B9FA 29                  ADD    HL,HL       ・(C, B) からIXにデータアドレス、HLに
B9FB 29                  ADD    HL,HL         表示アドレスを求める
B9FC 29                  ADD    HL,HL
B9FD 54                  LD     D,H
B9FE 5D                  LD     E,L
B9FF 29                  ADD    HL,HL
BA00 29                  ADD    HL,HL
BA01 19                  ADD    HL,DE
BA02 1110CA              LD     DE,DVRAM
BA05 19                  ADD    HL,DE
BA06 1600                LD     D,0
BA08 59                  LD     E,C
BA09 19                  ADD    HL,DE
BA0A C9                  RET

BA0B C5         MAPAD:   PUSH   BC
BA0C 3A82B7              LD     A,(MTOPX)
BA0F 81                  ADD    A,C
BA10 4F                  LD     C,A
BA11 3A83B7              LD     A,(MTOPY)
BA14 80                  ADD    A,B
BA15 47                  LD     B,A
BA16 2119BB              LD     HL,GDTOP-MXMAX  ・(C, B) からHLにマップデータ・アドレス
BA19 111800              LD     DE,MXMAX          を求める
BA1C 04                  INC    B
BA1D 19         MNOLP:   ADD    HL,DE
BA1E 10FD                DJNZ   MNOLP
BA20 09                  ADD    HL,BC
BA21 C1                  POP    BC
BA22 C9                  RET

BA23 6F         PTADR:   LD     L,A
BA24 07                  RLCA
BA25 85                  ADD    A,L
BA26 6F                  LD     L,A
BA27 2600                LD     H,0
BA29 29                  ADD    HL,HL       ・AからHLにパターンデータ・アドレスを
BA2A 29                  ADD    HL,HL         求める
BA2B 29                  ADD    HL,HL
BA2C 29                  ADD    HL,HL
BA2D 1141BA              LD     DE,GDATA
```

```
BA30 19                 ADD     HL,DE
BA31 C9                 RET

BA32 2190C7    BCADR:   LD      HL,DVRAM-80*8
BA35 118002             LD      DE,80*8
BA38 04                 INC     B
BA39 19        BCAL0:   ADD     HL,DE
BA3A 10FD               DJNZ    BCAL0          ・(C, B) からDEに表示アドレスを求める
BA3C CB21               SLA     C
BA3E 09                 ADD     HL,BC
BA3F EB                 EX      DE,HL
BA40 C9                 RET

BA41 00200200  GDATA:   DB      0,20H,2,0,40H,18H,4,0   ・パターンデータ
BA45 40180400
BA49 C8C01820           DB      0C8H,0C0H,18H,20H,40H,14H,1,090H
BA4D 40140190
BA51 65484830           DB      65H,48H,48H,30H,14H,090H,8,20H
BA55 14900820
BA59 00402002           DB      0,40H,20H,2,80H,42H,56H,0
BA5D 80425600
BA61 656ADBAA           DB      65H,6AH,0DBH,0AAH,62H,0DDH,66H,0DBH
BA65 62DD66DB
BA69 EDEBBAAA           DB      0EDH,0EBH,0BAH,0AAH,0E9H,097H,4DH,0BBH
BA6D E9974DBB

BA71 40008012           DB      40H,0,80H,12H,10H,0F9H,1,54H
BA75 10F90154
BA79 22AB0357           DB      22H,0ABH,3,57H,41H,0ACH,4,0FAH
BA7D 41AC04FA
BA81 0000001F           DB      0,0,0,1FH,0,0FBH,41H,5
BA85 00FB4105
BA89 02030203           DB      2,3,2,3,5,5,6,0FBH
BA8D 050506FB
BA91 F4C0DA00           DB      0F4H,0C0H,0DAH,0,0B8H,0F8H,51H,4
BA95 B8F85104
BA99 A2035203           DB      0A2H,3,52H,3,61H,4,60H,0F8H
BA9D 610460F8
BAA1 00004804           DB      0,0,48H,4,09FH,11H,2AH,88H
BAA5 9F112A88
BAA9 D540EAC0           DB      0D5H,40H,0EAH,0C0H,35H,80H,5FH,20H
BAAD 35805F20
BAB1 0000F800           DB      0,0,0F8H,0,0DFH,11H,0A0H,88H
BAB5 DF11A088
BAB9 C040C044           DB      0C0H,40H,0C0H,44H,0A0H,0A0H,0DFH,60H
BABD A0A0DF60
BAC1 0172002D           DB      1,72H,0,2DH,1FH,17H,20H,89H
BAC5 1F172089
BAC9 C04AC044           DB      0C0H,4AH,0C0H,44H,20H,85H,1FH,3
BACD 20851F03

BAD1 0492049A           DB      4,092H,4,09AH,9,1,8,0B0H
BAD5 090108B0
BAD9 13490011           DB      13H,49H,0,11H,2CH,0A0H,2,8
BADD 2CA00208
BAE1 0FFF1FDF           DB      0FH,0FFH,1FH,0DFH,3FH,0C1H,39H,0F0H
BAE5 3FC139F0
BAE9 35FF82BF           DB      35H,0FFH,82H,0BFH,80H,9,0,40H
BAED 80090040
BAF1 C0018000           DB      0C0H,1,80H,0,0,1,0,0
BAF5 00010000
BAF9 00008000           DB      0,0,80H,0,0C8H,2,0A9H,64H
BAFD C802A964

BB01 49205920           DB      49H,20H,59H,20H,80H,090H,9,8
BB05 80900908
BB09 91802E20           DB      091H,80H,2EH,20H,14H,0A8H,0,090H
BB0D 14A80090
BB11 FFF0FBF8           DB      0FFH,0F0H,0FBH,0F8H,83H,0FCH,0FH,09CH
BB15 83FC0F9C
```

```
BB19 FF24FAB0            DB      0FFH,24H,0FAH,0B0H,36H,8,0,10H
BB1D 36080010
BB21 80030001            DB      80H,3,0,1,80H,0,0,0
BB25 80000000
BB29 00080000            DB      0,8,0,0,0,0DH,42H,3BH
BB2D 000D423B

BB31 00000000  GDTOP:    DB      0,0,0,0,1,2,0,0,0,3,4,0,0,0,0,0,0,0,0,0,0,0,3,4
BB35 01020000
BB39 00030400                    ・マップデータ
BB3D 00000000
BB41 00000000
BB45 00000304
BB49 00000000            DB      0,0,0,0,3,4,0,0,0,0,0,0,0,1,2,0,0,0,1,2,0,0,0,0
BB4D 03040000
BB51 00000000
BB55 00010200
BB59 00000102
BB5D 00000000
BB61 00010200            DB      0,1,2,0,0,0,0,0,0,0,1,2,0,3,4,0,0,0,3,4,0,0,0,0
BB65 00000000
BB69 00000102
BB6D 00030400
BB71 00000304
BB75 00000000
BB79 00030400            DB      0,3,4,0,0,0,0,1,2,0,3,4,0,0,0,1,2,0,0,0,0,1,2,0
BB7D 00000001
BB81 02000304
BB85 00000001
BB89 02000000
BB8D 00010200
BB91 00000000            DB      0,0,0,0,0,0,0,3,4,0,0,0,0,0,0,3,4,0,0,0,0,3,4,0
BB95 00000003
BB99 04000000
BB9D 00000003
BBA1 04000000
BBA5 00030400
BBA9 00000001            DB      0,0,0,1,2,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
BBAD 02000000
BBB1 00000000
BBB5 00000000
BBB9 00000000
BBBD 00000000
BBC1 00000003            DB      0,0,0,3,4,0,0,0,0,0,0,1,2,0,0,0,0,0,0,0,0,0,0,0
BBC5 04000000
BBC9 00000001
BBCD 02000000
BBD1 00000000
BBD5 00000000
BBD9 00000000            DB      0,0,0,0,0,1,2,0,0,0,0,3,4,0,1,2,0,0,0,0,1,2,0,0
BBDD 00010200
BBE1 00000003
BBE5 04000102
BBE9 00000000
BBED 01020000
BBF1 01020000            DB      1,2,0,0,0,3,4,0,0,0,0,0,0,0,3,4,0,0,0,0,3,4,0,0
BBF5 00030400
BBF9 00000000
BBFD 00000304
BC01 00000000
BC05 03040000
BC09 03040000            DB      3,4,0,0,0,0,0,0,0,0,1,2,0,0,0,0,0,0,0,0,0,0,0,1
BC0D 00000000
BC11 00000102
BC15 00000000
BC19 00000000
BC1D 00000001
BC21 00000000            DB      0,0,0,0,0,0,0,0,0,0,3,4,0,0,0,0,0,1,2,0,0,0,0,3
BC25 00000000
BC29 00000304
```

```
BC2D 00000000
BC31 00010200
BC35 00000003
BC39 00000102          DB      0,0,1,2,0,0,0,0,1,2,0,0,0,0,0,0,0,3,4,0,0,0,0,0
BC3D 00000000
BC41 01020000
BC45 00000000
BC49 00030400
BC4D 00000000
BC51 00000304          DB      0,0,3,4,0,0,0,0,3,4,0,0,0,0,0,0,0,0,0,0,0,1,2,0
BC55 00000000
BC59 03040000
BC5D 00000000
BC61 00000000
BC65 00010200
BC69 00000000          DB      0,0,0,0,0,0,0,0,0,0,0,0,0,1,2,0,0,0,0,0,0,3,4,0
BC6D 00000000
BC71 00000000
BC75 00010200
BC79 00000000
BC7D 00030400
BC81 00000000          DB      0,0,0,0,1,2,0,0,0,0,0,0,0,3,4,0,0,0,0,1,2,0,0,0
BC85 01020000
BC89 00000000
BC8D 00030400
BC91 00000001
BC95 02000000
BC99 02000000          DB      2,0,0,0,3,4,0,0,0,1,2,0,0,0,0,0,1,2,0,3,4,0,0,0
BC9D 03040000
BCA1 00010200
BCA5 00000000
BCA9 01020003
BCAD 04000000
BCB1 04000000          DB      4,0,0,0,0,0,0,0,0,3,4,0,1,2,0,0,3,4,0,0,0,0,0,1
BCB5 00000000
BCB9 00030400
BCBD 01020000
BCC1 03040000
BCC5 00000001
```

芝生の中から無数のモグラが顔を出したような画面が表示されましたね。この時の表示は上から順にパターンを並べたものですが、見るからに平凡です。実は、これは平凡さを確認すると同時に、この画面からベタ・データを読んでいるのです。つまり、このデータをタイトルなどへの化粧表示へ利用しようというのです。化粧したほうの表示／消去はテンキーの[0][1][2]が対応しています。それぞれの内容は次のようなものです。なお、スペースキーを押している間はウェイトがかかるようになっていますから、実行過程をじっくりと見つめながら研究することもできます。

[0]：渦巻型の表示／消去（表示はマップパターンによる）

特にテクニックというものはありませんが、パターンを表示する順番におしゃれをしたものです。プログラムが面倒な割にはハデさはなく、おまけに表示に要する時間も結構長いため（ウェイトがないと渦巻に見えない）、何度も繰り返されるとカッタルイかもしれません。

[1]：ジワジワ表示／消去（表示はマップパターンによる）

画面を全体的にジワッと表示しジワッと消去するのは、ハイセンスなおしゃれといえます。おそらく画面を見ただけでは、表示／消去ともにアルゴリ

ズムがわからないでしょう。実は、表示は各パターンを1バイト（B／R／G）ずつ計16回に分けて表示しているのです。それが全体的にジワッと見える要因なのですが、その1バイトを単に上から順番に表示したのでは、ラインごとに表示したようにしか見えません。そのため、2バイト×8ドットのパターンを図3-4-2のような順で表示し、規則的にバラつかせているのです。

①	⑭
⑪	⑧
⑤	②
⑮	⑫
⑨	⑥
③	⑯
⑬	⑩
⑦	④

図3-4-2　ジワジワ表示の順序

一方、消去のほうは1バイトを2ドットずつ計4回で消去していますが、これも1ラインおきに消去の向きを変えています。図3-4-3を見て確認してください。正体がわかれば、どうということはないテクニックでしょう。

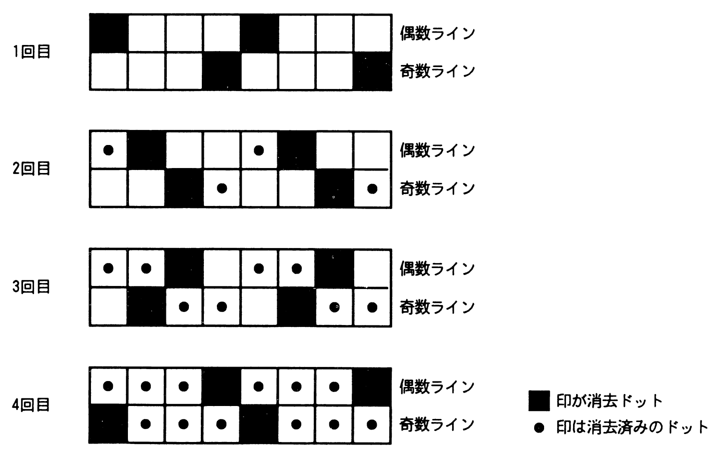


図3-4-3　ジワジワ消去の順序

2：斜め表示／消去（表示はベタ・データによる）

考え方としては、これもジワッと表示させるための1つの手段です。プレーンごとに斜めに表示（消去）していき、画面エンドになったら再びその座標をゼロに戻してループさせ、表示（消去）が全体的に行われるようにするものです。これは、3面同時に表示（消去）するよりも、プレーンごとに実行したほうがキラキラしてきれいに見えます。ここでは、ループの手順として、X方向が右エンドになったらY座標は変化させずにX座標だけを0にしていますが、場合によってはY座標も変化させなければならないことがあり

ます。例えば、図3-4-4の①ではどちらでもOKですが、②は前者の方法、③は後者の方法でないと同じ座標に戻ってしまいうまくいきません。

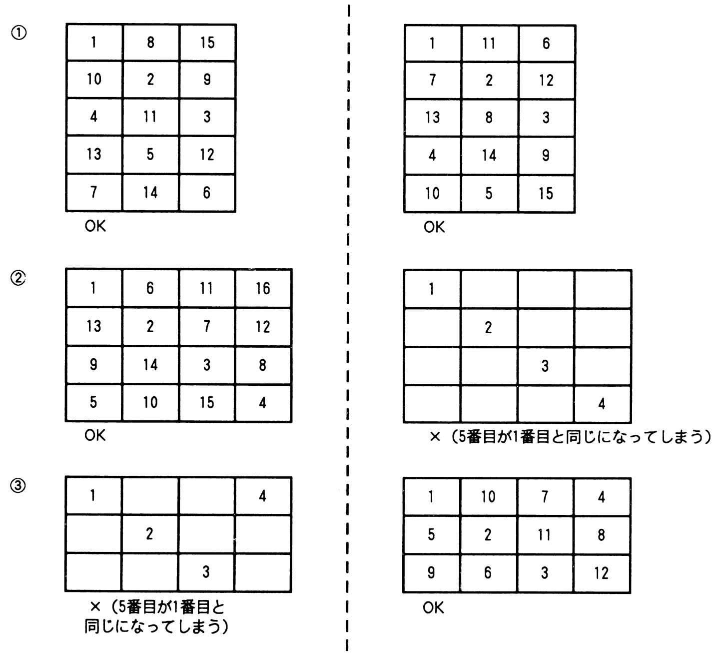


図3-4-4　斜め表示のアルゴリズムによる違い

この区別はプログラムで判断するよりも、やってみてダメならプログラムを手直しするという方法が最も簡単です。プログラム（LIST 3.14）中のラベル（D2COK／C2COK）の上の行（JR CGTHD／JR CV2AD）をカットすれば、後者のアルゴリズムになりますので、サイズを変えながら実験してみてください。

今回は消去にALUを用いませんでしたが、ALUのビットリセット・モードが消去には最適です。ドット単位で消去しながら1行おきに3面同時転送で左右にスクロールアウトするとか、フェードアウトのイメージと拡張グラフィックスの機能が合体すると、より美しいおしゃれができるかもしれません。

また、おしゃれの陰に隠れて目立ちませんが、このプログラムにはアセンブラの基礎化粧ともいうべきテクニックが潜んでいます。それは、ラベルの用法についてです。表示位置（DVRAM）、画面サイズ（GXMAX／GYMAX）、マップサイズ（MXMAX／MYMAX）はラベルで定義されていますが、これに関係するプログラム中の数値もすべてこのラベルによる計算式となっています。そのため、表示位置や各サイズの変更は定義されたラベルの値を変更するだけで済み、プログラム本体はまったくいじる必要がありません。ゲーム制作途中で画面デザインを変えたい……というようなことでバグなど出さぬよう、こんな基礎化粧品があることも忘れてはなりません。

3-5 漢字処理のすべて

ひらがなの「び」からは「び」以外のことは伝わってきませんが、漢字で「美」と書いてあれば「うつくしい」というイメージも伝わってきます。文字を読むことで物語が進行する小説やゲームにおいては、イメージの伝播というものは大変重要です。「びぶん」では微分か美文かわかりませんからね。

初期のゲームにおいては、文字といえばテキスト画面のカタカナ文字でした。それが文字フォントを用意したひらがな／カタカナ表示となり、いまではほとんどのゲームで漢字が使われるようになっています。PC-8801mk II以降は漢字ROMが標準で装備されているのですから、当然といえば当然なのですが、漢字ROMのフォントサイズが16×16ドット（全角の場合）で作られているため、カラーグラフィック画面で表示しようとすると縦長の大きな文字になってしまいます。そのため、これを1/2（あるいは3/4や11/16など）に縦縮小して表示するゲームが多くなっています。さらには、特徴を出すために1ビット横にずらして重ね合わせる（文字を太くする）とか、カッコよく斜体文字にするなどおしゃれなテクニックが見られることもあります。

それでは、まず漢字ROMに収められているフォント（データ）はどのような構造になっているのか、そしてどのようにしてリードするのかを解説することにしましょう。

本体付属のマニュアルには漢字コード表が資料として載っていますが、漢字（記号等も含む）にはすべてJISで定められたコード番号がつけられています。一方、漢字ROMというのは文字フォントが収められたメモリであり、1つの文字を表すために複数のメモリが使われています。当然、漢字JISコードと漢字ROMアドレスは一致しておらず、最初にこの変換作業を行わなければなりません。

ところで、文字には全角文字、半角文字、1/4角文字の3種類があり、それぞれのフォントデータは次のような順でメモリに格納されています。

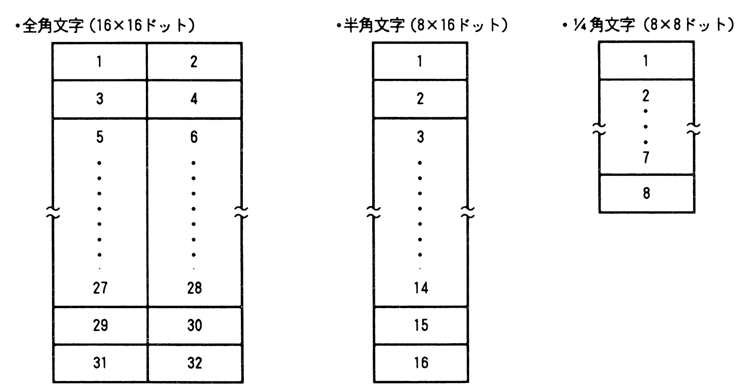


図3-5-1　文字データの格納順

そこで、漢字JISコードから漢字ROMアドレスへ変換する手順ですが、全角文字はさらに非漢字（記号やひらがな／カタカナ等）と漢字に分けられ、図3-5-2のように何ともややこしい変換をしなければなりません。

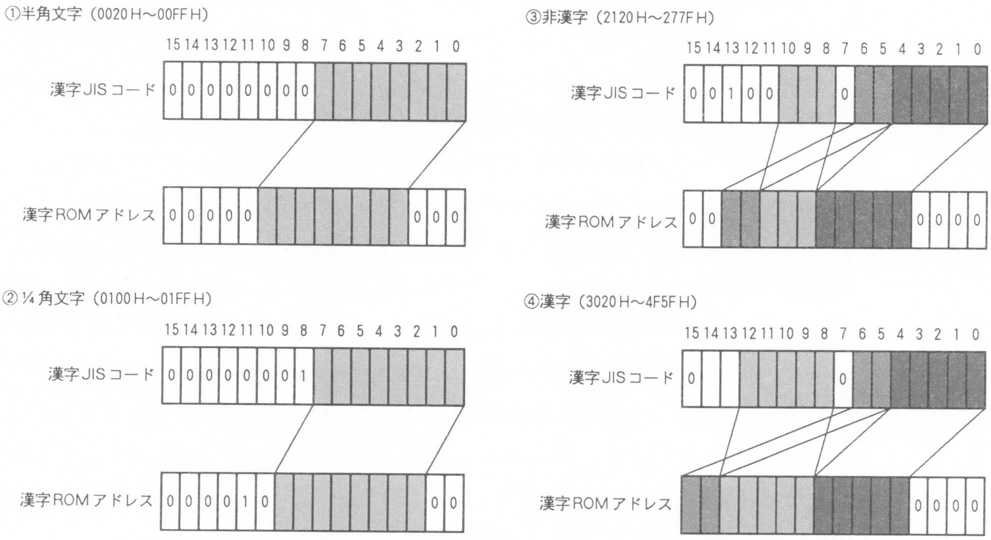


図3-5-2　漢字JISコード→漢字ROMアドレスへの変換（第一水準の場合）

面倒とはいえ、いったんプログラムにしてしまえば自動的に変換されるわけですから、特に悩むことはないでしょう。ただ、漢字の変換でビットの13、14が無視された形になっているのが気になるかもしれません。これは、第一水準の漢字ではJISコードの上位4ビットが3または4ですから、1ビットあれば区別できるためです。

```
3XXXH = 0011XXXXXXXXXXXXB
4XXXH = 0100XXXXXXXXXXXXB
```

ゲームで使うことはないと思いますが、参考までに第二水準漢字の変換手順も示しておきます（図3-5-3）。ただし、第二水準漢字の場合は漢字ROMアドレスを出力するポートが違いますから気を付けてください（図3-5-4の注参照）。

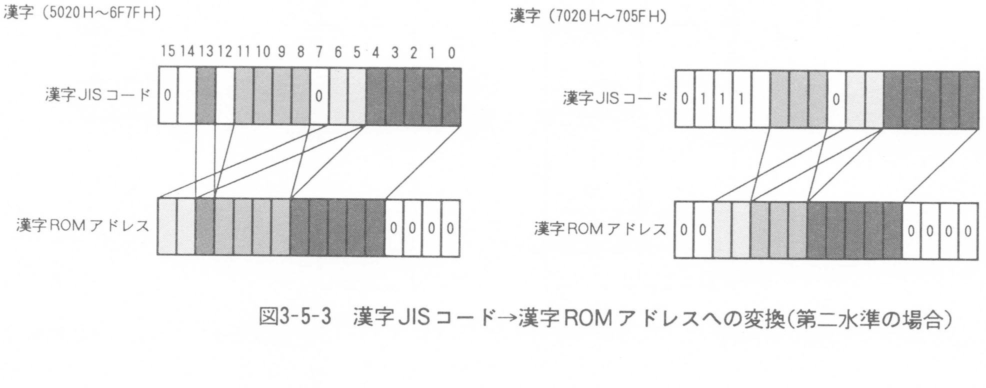
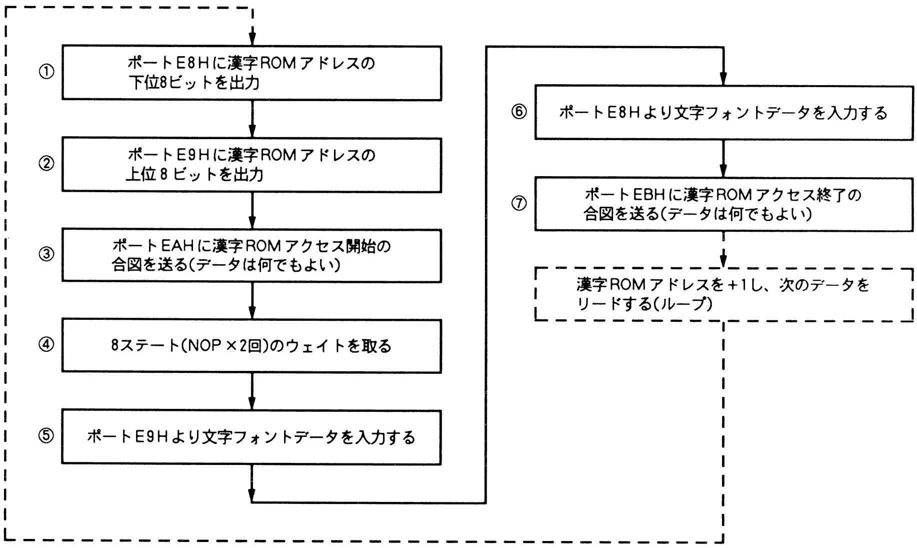


図3-5-3　漢字JISコード→漢字ROMアドレスへの変換（第二水準の場合）

次に、漢字ROMアドレスから実際にデータをリードする方法ですが、これにはI／OポートのE8H～EBHを使用します。具体的には図3-5-4のような流れでリードすることになります。1つのアドレスから2バイトのデータを得られますから、全角文字の場合のループ回数は16回です。なお、図中③④⑦の項目はPC-8801FR以降は省略できますが、全機種対応にするためには入れておく必要があります。さらに、⑦については漢字ROMアクセス前にダミーとして1回は実行しなければなりません。ゲームのスタート時に、無条件で実行しておくといいでしょう。



注）第二水準の漢字の場合は、E8H→ECH、E9H→EDHとなる。

図3-5-4　データ・リードのフロー(第一水準漢字の場合)

ここまでが、漢字ROM操作の基礎の部分です。このデータをどう加工するか、ここで縦縮小表示と斜体表示の実体について簡単に触れておきます。縦縮小というのは、16ライン分のデータをどこかでダブらせて（ORで重ね合わせて）表示し、全体のライン数を減らすものです。1ケ所ダブらせれば1ライン少なくなるわけですが、美しい縮小文字にするためには、どのラインをダブらせるかがポイントです。例えば、4ケ所をダブらせる3/4縮小にしても、単純に4ライン単位でダブらせたのでは美しい縮小文字にはなりません。文字の種類（記号、ひらがな、漢字……等）による変化、さらには例外処理をも考慮することによって、初めて縮小を感じさせない縮小文字が得られるのです。「雰囲気で読めればいい」という1/2縮小や、単純一律な縮小を採用したゲームもありますが、文字本来の美しさを無視するようで個人的に好きではありません。

一方、縮小がデータの縦変形なら、斜体文字はデータの横変形といったものです。つまり、データを左右にシフトして表示するわけですが、こういったものはアイデア次第でいくらでも考えられます。参考として、図3-5-5に「愛」を斜体にした例を示しますが、これにとらわれずに色々なデータ変形にトライしてください。縮小は縦200ラインという制限を補うためのものですが、それ以外の変形はオ

リジナリティが勝負ですから。

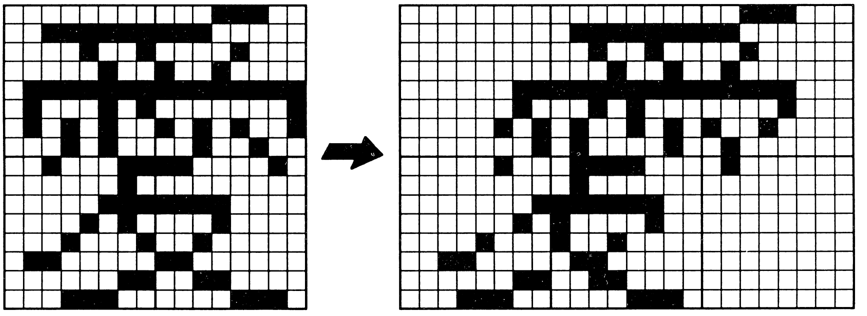


図3-5-5　斜体文字の例

では、最後に11/16縦縮小による本格的文字列表示ルーチンを紹介します。ゲーム用ということで第一水準の全角文字にしか対応していませんが、ダブらせるラインに関しては文字の美形を損なわないためにかなり気を配っています。主要記号／英大文字／数字／ひらがな／カタカナについては、例外も含めてすべて自動的に最適の縮小がなされ、また漢字については最も汎用的な縮小を標準とし、合わない場合にはマニュアル処理もできるようになっています。

なお、このプログラム（LIST 3.15）は表示にALUを用いているため、PC-8801SR以降専用です。

```
LINE(0,0)-(367,146),7,B
CLEAR,&HB5FF:CLS:A=&HB700:CALL A
```

LIST 3.15

```
                        ORG     0B700H

0000=           @END:   EQU     0
0001=           @WT:    EQU     1
0002=           @MAX:   EQU     2
0003=           @NEX:   EQU     3
0004=           @SUB:   EQU     4
0005=           @DJP:   EQU     5

0006=           @HI:    EQU     6
0007=           @KA:    EQU     7
0008=           @EI:    EQU     8
0009=           @GI:    EQU     9
000A=           @RO:    EQU     10

000B=           @NY:    EQU     11
000C=           _:      EQU     12
000D=           __:     EQU     13
000E=           、:     EQU     14
000F=           •:      EQU     15
0010=           .:      EQU     16
0011=           ?:      EQU     17
0012=           !:      EQU     18
0013=           」:     EQU     19
0014=           」」:   EQU     20
0015=           @」:    EQU     21
0016=           -:      EQU     22
0017=           ・:     EQU     23
```

```
0018=          ･･･:    EQU     24
0019=          ｢:      EQU     25
001A=          ｢｢:     EQU     26
001B=          @｢:     EQU     27

001C=          @KI1:   EQU     28
001D=          @KI2:   EQU     29
001E=          @SPK:   EQU     30
001F=          @GAI:   EQU     31

00A1=          ｧ:      EQU     0A1H
00A2=          ｱ:      EQU     0A2H
00A3=          ｨ:      EQU     0A3H
00A4=          ｲ:      EQU     0A4H
00A5=          ｩ:      EQU     0A5H
00A6=          ｳ:      EQU     0A6H
00A7=          ｪ:      EQU     0A7H
00A8=          ｴ:      EQU     0A8H
00A9=          ｫ:      EQU     0A9H
00AA=          ｵ:      EQU     0AAH
00AB=          ｶ:      EQU     0ABH
00AC=          ｶﾞ:     EQU     0ACH
00AD=          ｷ:      EQU     0ADH
00AE=          ｷﾞ:     EQU     0AEH
00AF=          ｸ:      EQU     0AFH

00B0=          ｸﾞ:     EQU     0B0H
00B1=          ｹ:      EQU     0B1H
00B2=          ｹﾞ:     EQU     0B2H
00B3=          ｺ:      EQU     0B3H
00B4=          ｺﾞ:     EQU     0B4H
00B5=          ｻ:      EQU     0B5H
00B6=          ｻﾞ:     EQU     0B6H
00B7=          ｼ:      EQU     0B7H
00B8=          ｼﾞ:     EQU     0B8H          ・文字列用のラベル
00B9=          ｽ:      EQU     0B9H
00BA=          ｽﾞ:     EQU     0BAH
00BB=          ｾ:      EQU     0BBH
00BC=          ｾﾞ:     EQU     0BCH
00BD=          ｿ:      EQU     0BDH
00BE=          ｿﾞ:     EQU     0BEH
00BF=          ﾀ:      EQU     0BFH

00C0=          ﾀﾞ:     EQU     0C0H
00C1=          ﾁ:      EQU     0C1H
00C2=          ﾁﾞ:     EQU     0C2H
00C3=          ｯ:      EQU     0C3H
00C4=          ﾂ:      EQU     0C4H
00C5=          ﾂﾞ:     EQU     0C5H
00C6=          ﾃ:      EQU     0C6H
00C7=          ﾃﾞ:     EQU     0C7H
00C8=          ﾄ:      EQU     0C8H
00C9=          ﾄﾞ:     EQU     0C9H
00CA=          ﾅ:      EQU     0CAH
00CB=          ﾆ:      EQU     0CBH
00CC=          ﾇ:      EQU     0CCH
00CD=          ﾈ:      EQU     0CDH
00CE=          ﾉ:      EQU     0CEH
00CF=          ﾊ:      EQU     0CFH

00D0=          ﾊﾞ:     EQU     0D0H
00D1=          ﾊﾟ:     EQU     0D1H
00D2=          ﾋ:      EQU     0D2H
00D3=          ﾋﾞ:     EQU     0D3H
00D4=          ﾋﾟ:     EQU     0D4H
00D5=          ﾌ:      EQU     0D5H
00D6=          ﾌﾞ:     EQU     0D6H
00D7=          ﾌﾟ:     EQU     0D7H
00D8=          ﾍ:      EQU     0D8H
00D9=          ﾍﾞ:     EQU     0D9H
```

```
00DA=          ヘ゜:     EQU     0DAH
00DB=          ホ:       EQU     0DBH
00DC=          ホ゛:     EQU     0DCH
00DD=          ホ゜:     EQU     0DDH
00DE=          マ:       EQU     0DEH
00DF=          ミ:       EQU     0DFH

00E0=          ム:       EQU     0E0H
00E1=          メ:       EQU     0E1H
00E2=          モ:       EQU     0E2H
00E3=          ャ:       EQU     0E3H
00E4=          ヤ:       EQU     0E4H
00E5=          ュ:       EQU     0E5H
00E6=          ユ:       EQU     0E6H
00E7=          ョ:       EQU     0E7H
00E8=          ヨ:       EQU     0E8H
00E9=          ラ:       EQU     0E9H
00EA=          リ:       EQU     0EAH
00EB=          ル:       EQU     0EBH
00EC=          レ:       EQU     0ECH
00ED=          ロ:       EQU     0EDH
00EF=          ワ:       EQU     0EFH

00F2=          ヲ:       EQU     0F2H
00F3=          ン:       EQU     0F3H
00F4=          ウ゛:     EQU     0F4H

00F8=          C0:       EQU     0F8H
00F9=          C1:       EQU     0F9H
00FA=          C2:       EQU     0FAH
00FB=          C3:       EQU     0FBH
00FC=          C4:       EQU     0FCH
00FD=          C5:       EQU     0FDH
00FE=          C6:       EQU     0FEH
00FF=          C7:       EQU     0FFH

B700 F3        KTEST:    DI                                  ・割り込み禁止
B701 ED7314B7            LD      (SPPOP+1),SP               ┐
B705 3100B7              LD      SP,0B700H                  ┘・SPの退避と設定
B708 D3EB                OUT     (0EBH),A                    ・漢字ROM使用の前に一度は実行しておくこと
B70A 2118B7              LD      HL,KDATA                    ・HL＝文字列の先頭アドレス
B70D 11F2C0              LD      DE,0C000H+80*3+2            ・DE＝表示位置
B710 CD4DB8              CALL    KAPUT                       ・文字列の表示
B713 310000    SPPOP:    LD      SP,0                        ・SPを元に戻す
B716 FB                  EI                                  ・割込み許可
B717 C9                  RET                                 ・BASICへ戻る

B718 0101022A  KDATA:    DB      @WT,1,@MAX,42,@NEX    ・サンプル文字列データ
B71C 03
B71D 1004                DW      80*13
B71F 0DFE04              DB      __,C6,@SUB
B722 F3B7                DW      KDSUB
B724 36584227            DB      36H,58H,42H,27H,3DH,68H,4DH,7DH
B728 3D684D7D
B72C E2B3CEE8            DB      モ,コ,ノ,ヨ,ウ,ニ,39H,54H,ワ,レ,マ,ス,。
B730 A6CB3954
B734 EFECDEB9
B738 0F
B739 DEBF0E33            DB      マ,タ,、,33H,30H,3BH,7AH,ヲ,3BH,48H,ウ
B73D 303B7AF2
B741 3B48A6
B744 B3C8E2C7            DB      コ,ト,モ,デ゛,キ,マ,ス,。,@NY
B748 ADDEB90F
B74C 0B
B74D FD1B4E63            DB      C5,@「,4EH,63H,@」,・・・,・・・,_,@GAI
B751 1518180C
B755 1F
B756 2DB8                DW      GASMP
B758 0B0DFA              DB      @NY,__,C2
B75B 3C21CB0E            DB      3CH,21H,ニ,、,@KA,サ,ン,プ゜,ル,@HI,ト,シ,テ,@SUB
```

```
B75F 07B5F3D7
B763 EB06C8B7
B767 C604
B769 19B8                DW      SPCEI
B76B 3F743B7A            DB      3FH,74H,3BH,7AH,ｦ,@SUB
B76F F204
B771 28B8                DW      HYOJI
B773 B7C6DFDE            DB      ｼ,ﾃ,ﾐ,ﾏ,ｼ,ｮ,ｳ,｡,@NY,C4,｢｢,@EI
B777 B7E7A60F
B77B 0BFC1A08
B77F 41424344            DB      'ABCDEFGHIJKLMNOPQRSTUVWXYZ'
B783 45464748
B787 494A4B4C
B78B 4D4E4F50
B78F 51525354
B793 55565758
B797 595A
B799 0DF93031            DB      __,C1,'0123456789',C4,｣｣,@HI
B79D 32333435
B7A1 36373839
B7A5 FC1406
B7A8 0BFE1C79            DB      @NY,C6,@KI1,79H,_,C7,｢,@SUB
B7AC 0CFF1904
B7B0 19B8                DW      SPCEI
B7B2 131D2B0C            DB      ｣,@KI2,2BH,_,46H,43H,4AH,4CH,@SUB
B7B6 46434A4C
B7BA 04
B7BB 23B8                DW      SYUKU
B7BD C7B90F07            DB      ﾃﾞ,ｽ,｡,@KA,ﾉ,ｰ,ﾏ,ﾙ,@SUB
B7C1 CE16DEEB
B7C5 04
B7C6 23B8                DW      SYUKU
B7C8 06C0C819            DB      @HI,ﾀﾞ,ﾄ,｢,31H,51H,｣,ﾄ,ﾅ,ﾘ,ﾏ,ｽ,｡
B7CC 315113C8
B7D0 CAEADEB9
B7D4 0F
B7D5 0BFB063B            DB      @NY,C3,@HI,3BH,48H,ｲ,4AH,7DH,ﾊ,･,･,･,､
B7D9 48A44A7D
B7DD CF171717
B7E1 0E
B7E2 4B5C4A38            DB      4BH,5CH,4AH,38H,ｦ,ﾖ,ｸ
B7E6 F2E8AF
B7E9 4649F3C7            DB      46H,49H,ﾝ,ﾃﾞ,ｸ,ﾀﾞ,ｻ,ｲ,｡,@END
B7ED AFC0B5A4
B7F1 0F00

B7F3 06B3ECCF  KDSUB:    DB      @HI,ｺ,ﾚ,ﾊ,3DH,44H,@EI,'11',@KA,ﾗ,ｲ,ﾝ,@HI,ﾆ,@SUB
B7F7 3D440831                                          ・サンプル文字列のサブデータ
B7FB 3107E9A4
B7FF F306CB04
B803 23B8                DW      SYUKU
B805 B5ECBF4A            DB      ｻ,ﾚ,ﾀ,4AH,38H,3BH,7AH,@SUB
B809 383B7A04
B80D 28B8                DW      HYOJI
B80F 07EB16C1            DB      @KA,ﾙ,ｰ,ﾁ,ﾝ,@HI,ﾃﾞ,ｽ,｡,@END
B813 F306C7B9
B817 0F00

B819 1E315101  SPCEI:    DB      @SPK,31H,51H,1,3,3,4,4,1,@END  ・特別縮小漢字の例
B81D 03030404
B821 0100
B823 3D4C3E2E  SYUKU:    DB      3DH,4CH,3EH,2EH,@END           ・漢字コード「縮小」
B827 00
B828 493D3C28  HYOJI:    DB      49H,3DH,3CH,28H,@END           ・漢字コード「表示」
B82C 00

B82D 07E0      GASMP:    DB      00000111B,11100000B
B82F 1818                DB      00011000B,00011000B
B831 300C                DB      00110000B,00001100B
B833 6666                DB      01100110B,01100110B
```

```
B835 6666                 DB     01100110B,01100110B    ・サンプル外字
B837 6006                 DB     01100000B,00000110B
B839 6C36                 DB     01101100B,00110110B
B83B 67E6                 DB     01100111B,11100110B
B83D 300C                 DB     00110000B,00001100B
B83F 0FF0                 DB     00001111B,11110000B
B841 781E                 DB     01111000B,00011110B

B843 C5        WAIT:      PUSH   BC                     ・ウェイト・ルーチン
B844 0E64                 LD     C,100
B846 10FE      WAITL:     DJNZ   WAITL
B848 0D                   DEC    C
B849 20FB                 JR     NZ,WAITL
B84B C1                   POP    BC
B84C C9                   RET

B84D DB32      KAPUT:     IN     A,(32H)                ・文字列表示ルーチン・エントリー
B84F F640                 OR     01000000B                拡張アクセスモード
B851 D332                 OUT    (32H),A                  カラー＝7
B853 3E07                 LD     A,7                      ジャンプ・ポインタ初期化
B855 3240B9    KAPCX:     LD     (RCCOL+1),A
B858 CD71B8    KAPDI:     CALL   OUT34
B85B 016CBA               LD     BC,KANDP
B85E ED4323B9             LD     (KAPON+1),BC
B862 011ABA               LD     BC,HIDIP
B865 ED4329B9             LD     (MOJPO+1),BC
B869 D5                   PUSH   DE
B86A D9                   EXX
B86B D1                   POP    DE
B86C 0E00                 LD     C,0
B86E C3C0B8               JP     CKCOM

B871 D334      OUT34:     OUT    (34H),A                ・カラー変化とG.VRAMセレクト
B873 3E80                 LD     A,10000000B
B875 D335                 OUT    (35H),A
B877 C9                   RET

B878 D9        LFEED:     EXX                           ・ラインフィード
B879 EB                   EX     DE,HL
B87A 011004    LFLEN:     LD     BC,80*13
B87D 09                   ADD    HL,BC
B87E EB                   EX     DE,HL
B87F D5                   PUSH   DE
B880 D9                   EXX
B881 D1                   POP    DE
B882 0E00                 LD     C,0
B884 C9                   RET

B885 AF        KDA00:     XOR    A                      ・独立アクセスモード／メインメモリ・セレクト
B886 D335                 OUT    (35H),A
B888 DB32                 IN     A,(32H)
B88A E6BF                 AND    10111111B
B88C D332                 OUT    (32H),A
B88E C9                   RET

B88F D9        KDA01:     EXX                           ・ウェイト値セット
B890 7E                   LD     A,(HL)
B891 23                   INC    HL
B892 32BBB8               LD     (WTCHK+1),A
B895 182A                 JR     CKOM1

B897 D9        KDA02:     EXX                           ・コマ数セット
B898 7E                   LD     A,(HL)
B899 23                   INC    HL
B89A 3206B9               LD     (KOMA2+1),A
B89D C602                 ADD    A,2
B89F 32FAB8               LD     (KOMA1+1),A
B8A2 181D                 JR     CKOM1

B8A4 D9        KDA03:     EXX
B8A5 4E                   LD     C,(HL)
B8A6 23                   INC    HL
```

```
B8A7 46                 LD      B,(HL)           ・改行ライン数セット
B8A8 23                 INC     HL
B8A9 ED437BB8           LD      (LFLEN+1),BC
B8AD 1812               JR      CKOM1

B8AF CD78B8   NLINE:    CALL    LFEED            ・改行
B8B2 180C               JR      CKCOM

B8B4 13       INCC2:    INC     DE               ・2スペース
B8B5 0C                 INC     C
B8B6 13       INCC1:    INC     DE               ・1スペース
B8B7 0C                 INC     C
B8B8 1806               JR      CKCOM

B8BA 3E00     WTCHK:    LD      A,0              ・表示ウェイトのチェック
B8BC B7                 OR      A
B8BD C443B8             CALL    NZ,WAIT
B8C0 D9       CKCOM:    EXX
B8C1 7E       CKOM1:    LD      A,(HL)
B8C2 23                 INC     HL
B8C3 D9                 EXX
B8C4 FEF8               CP      0F8H
B8C6 380A               JR      C,CKCO1
B8C8 D6F8               SUB     0F8H
B8CA 3240B9             LD      (RCCOL+1),A
B8CD CD71B8             CALL    OUT34
B8D0 18EE               JR      CKCOM
B8D2 FE01     CKCO1:    CP      1
B8D4 38AF               JR      C,KDA00
B8D6 28B7               JR      Z,KDA01
B8D8 FE03               CP      3
B8DA 38BB               JR      C,KDA02
B8DC 28C6               JR      Z,KDA03
B8DE FE05               CP      5
B8E0 DA2BB9             JP      C,RECAL
B8E3 CA47B9             JP      Z,DAJMP
B8E6 FE0B               CP      11
B8E8 DA59B9             JP      C,MOJCC
B8EB 28C2               JR      Z,NLINE
B8ED FE0D               CP      13
B8EF 38C5               JR      C,INCC1
B8F1 28C1               JR      Z,INCC2
B8F3 21BAB8             LD      HL,WTCHK         ・文字列表示メインルーチン
B8F6 E5                 PUSH    HL
B8F7 6F                 LD      L,A
B8F8 79                 LD      A,C
B8F9 FE2C     KOMA1:    CP      44
B8FB D478B8             CALL    NC,LFEED
B8FE 7D                 LD      A,L
B8FF FE16               CP      22
B901 DA89B9             JP      C,KINSO
B904 79                 LD      A,C
B905 FE2A     KOMA2:    CP      42
B907 D478B8             CALL    NC,LFEED
B90A 7D                 LD      A,L
B90B FE1C               CP      28
B90D DA90B9             JP      C,KDKIG
B910 CAA0B9             JP      Z,KI1DI
B913 FE1E               CP      30
B915 DAA4B9             JP      C,KI2DI
B918 CAC5B9             JP      Z,KDASP
B91B FE1F               CP      31
B91D CAD0B9             JP      Z,GAIJD
B920 FE80               CP      80H
B922 DA6CBA   KAPON:    JP      C,KANDP
B925 E67F               AND     01111111B
B927 6F                 LD      L,A
B928 C31ABA   MOJPO:    JP      HIDIP

B92B D9       RECAL:    EXX
B92C 4E                 LD      C,(HL)
```

```
B92D 23                 INC     HL
B92E 46                 LD      B,(HL)
B92F 23                 INC     HL
B930 E5                 PUSH    HL
B931 60                 LD      H,B
B932 69                 LD      L,C                 ・サブデータ処理
B933 D9                 EXX
B934 CDC0B8             CALL    CKCOM
B937 D9                 EXX
B938 E1                 POP     HL
B939 DB32               IN      A,(32H)
B93B F640               OR      01000000B
B93D D332               OUT     (32H),A
B93F 3E07     RCCOL:    LD      A,7
B941 CD71B8             CALL    OUT34
B944 C3C1B8             JP      CKOM1

B947 D9       DAJMP:    EXX
B948 7E                 LD      A,(HL)
B949 23                 INC     HL
B94A 66                 LD      H,(HL)              ・データアドレス変更
B94B 6F                 LD      L,A
B94C C3C1B8             JP      CKOM1
                                                           ・文字処理用ジャンプテーブル
B94F 1ABA2EBA MJPDT:    DW      HIDIP,KTDIP,ESDIP,GIDIP,RODIP
B953 F1B90BBA
B957 0FBA

B959 D606     MOJCC:    SUB     6
B95B FE02               CP      2
B95D 216CBA             LD      HL,KANDP
B960 3803               JR      C,MHIKA
B962 2128B9             LD      HL,MOJPO
B965 2223B9   MHIKA:    LD      (KAPON+1),HL
B968 07                 RLCA
B969 214FB9             LD      HL,MJPDT
B96C 85                 ADD     A,L
B96D 6F                 LD      L,A                 ・各文字処理ルーチンへジャンプ
B96E 3001               JR      NC,MOJC1
B970 24                 INC     H
B971 7E       MOJC1:    LD      A,(HL)
B972 23                 INC     HL
B973 66                 LD      H,(HL)
B974 6F                 LD      L,A
B975 2229B9             LD      (MOJPO+1),HL
B978 C3C0B8             JP      CKCOM
                                                    ・禁則処理のある特殊記号用コードデータ
B97B 22232529 KNSDT:    DB      22H,23H,25H,29H,2AH,57H,59H,4BH
B97F 2A57594B
B983 3C264456 KKIDT:    DB      3CH,26H,44H,56H,58H,4AH       ・特殊記号用コードデータ
B987 584A

B989 D60E     KINSO:    SUB     14
B98B 217BB9             LD      HL,KNSDT            ・特殊記号（禁則処理あり）表示
B98E 1805               JR      KDKI1
B990 D616     KDKIG:    SUB     22
B992 2183B9             LD      HL,KKIDT
B995 85       KDKI1:    ADD     A,L
B996 6F                 LD      L,A
B997 3001               JR      NC,KDKI2
B999 24                 INC     H                   ・特殊記号表示
B99A 7E       KDKI2:    LD      A,(HL)
B99B 6F                 LD      L,A
B99C 2621               LD      H,21H
B99E 180B               JR      KIGDI

B9A0 2621     KI1DI:    LD      H,21H               ・記号表示－①
B9A2 1802               JR      KDAKI
B9A4 2622     KI2DI:    LD      H,22H
B9A6 D9       KDAKI:    EXX
```

```
B9A7 7E                  LD      A,(HL)
B9A8 23                  INC     HL
B9A9 D9                  EXX
B9AA 6F                  LD      L,A
B9AB F5         KIGDI:   PUSH    AF          ・記号表示－②
B9AC CDFDBA              CALL    KIADR
B9AF F1                  POP     AF
B9B0 FE61                CP      61H
B9B2 CA75BA              JP      Z,KAND1
B9B5 FD21C7BA            LD      IY,KUTEN
B9B9 FE23                CP      23H
B9BB CA79BA              JP      Z,MODIP
B9BE FD21C1BA            LD      IY,KIGOU
B9C2 C379BA              JP      MODIP

B9C5 D9         KDASP:   EXX
B9C6 46                  LD      B,(HL)
B9C7 23                  INC     HL
B9C8 4E                  LD      C,(HL)
B9C9 23                  INC     HL          ・特別縮小漢字表示
B9CA C5                  PUSH    BC
B9CB D9                  EXX
B9CC E1                  POP     HL
B9CD C358BA              JP      SPKAN

B9D0 D9         GAIJD:   EXX
B9D1 4E                  LD      C,(HL)
B9D2 23                  INC     HL
B9D3 46                  LD      B,(HL)
B9D4 23                  INC     HL
B9D5 C5                  PUSH    BC
B9D6 D9                  EXX
B9D7 E1                  POP     HL
B9D8 D5                  PUSH    DE
B9D9 060B                LD      B,11
B9DB 7E         GAIDL:   LD      A,(HL)      ・外字表示
B9DC 12                  LD      (DE),A
B9DD 23                  INC     HL
B9DE 13                  INC     DE
B9DF 7E                  LD      A,(HL)
B9E0 12                  LD      (DE),A
B9E1 23                  INC     HL
B9E2 3E4F                LD      A,79
B9E4 CDECB9              CALL    ADEAA
B9E7 10F2                DJNZ    GAIDL
B9E9 C39EBA              JP      MDPOP

B9EC 83         ADEAA:   ADD     A,E
B9ED 5F                  LD      E,A
B9EE D0                  RET     NC          ・DE＝DE＋A
B9EF 14                  INC     D
B9F0 C9                  RET

B9F1 2623       ESDIP:   LD      H,23H
B9F3 FE39                CP      39H
B9F5 280B                JR      Z,SUJI9
B9F7 3018                JR      NC,ESABC    ・英数字表示
B9F9 CDFDBA              CALL    KIADR
B9FC FD21D3BA            LD      IY,SUJ08
BA00 1877                JR      MODIP

BA02 CDFDBA     SUJI9:   CALL    KIADR
BA05 FD21D9BA            LD      IY,SUJD9    ・数字「9」表示
BA09 186E                JR      MODIP

BA0B 2626       GIDIP:   LD      H,26H       ・ギリシア文字表示
BA0D 1802                JR      ESABC
BA0F 2627       RODIP:   LD      H,27H
BA11 CDFDBA     ESABC:   CALL    KIADR       ・ロシア文字表示
BA14 FD21CDBA            LD      IY,EIMOJ
BA18 185F                JR      MODIP
```

```
BA1A 2624      HIDIP:  LD      H,24H        ・ひらがな表示
BA1C FE67              CP      67H
BA1E 2826              JR      Z,SPHI2
BA20 D62F              SUB     2FH
BA22 FE02              CP      2
BA24 3817              JR      C,SPHI1
BA26 D606              SUB     35H-2FH
BA28 FE06              CP      6
BA2A 381A              JR      C,SPHI2
BA2C 1806              JR      GETNK
BA2E 2625      KTDIP:  LD      H,25H        ・カタカナ表示
BA30 FE26              CP      26H
BA32 281B              JR      Z,SPKA1
BA34 CDFDBA    GETNK:  CALL    KIADR
BA37 FD21DFBA          LD      IY,KAHIR
BA3B 183C              JR      MODIP

BA3D CDFDBA    SPHI1:  CALL    KIADR        ・「く」「ぐ」表示
BA40 FD21E5BA          LD      IY,HIGA1
BA44 1833              JR      MODIP

BA46 CDFDBA    SPHI2:  CALL    KIADR        ・「さ」「し」「す」表示
BA49 FD21EBBA          LD      IY,HIGA2
BA4D 182A              JR      MODIP

BA4F CDFDBA    SPKA1:  CALL    KIADR        ・「ウ」表示
BA52 FD21F1BA          LD      IY,KATA1
BA56 1821              JR      MODIP

BA58 CD08BB    SPKAN:  CALL    KAADR        ・特別縮小漢字表示
BA5B D5                PUSH    DE
BA5C D9                EXX
BA5D 0606              LD      B,6
BA5F 7E        SKADL:  LD      A,(HL)
BA60 23                INC     HL
BA61 D9                EXX
BA62 47                LD      B,A
BA63 CDA4BA            CALL    MODI2
BA66 D9                EXX
BA67 10F6              DJNZ    SKADL
BA69 D9                EXX
BA6A 1832              JR      MDPOP

BA6C 67        KANDP:  LD      H,A          ・文字（漢字）表示
BA6D D9                EXX
BA6E 7E                LD      A,(HL)
BA6F 23                INC     HL
BA70 D9                EXX
BA71 6F                LD      L,A
BA72 CD08BB            CALL    KAADR
BA75 FD21F7BA  KAND1:  LD      IY,KANJI
BA79 D5        MODIP:  PUSH    DE
BA7A FD4600            LD      B,(IY+0)
BA7D CDA4BA            CALL    MODI2
BA80 FD4601            LD      B,(IY+1)
BA83 CDA4BA            CALL    MODI2
BA86 FD4602            LD      B,(IY+2)
BA89 CDA4BA            CALL    MODI2
BA8C FD4603            LD      B,(IY+3)
BA8F CDA4BA            CALL    MODI2
BA92 FD4604            LD      B,(IY+4)
BA95 CDA4BA            CALL    MODI2
BA98 FD4605            LD      B,(IY+5)
BA9B CDA4BA            CALL    MODI2
BA9E D1        MDPOP:  POP     DE
BA9F 13                INC     DE
BAA0 13                INC     DE
BAA1 0C                INC     C
BAA2 0C                INC     C
BAA3 C9                RET
```

```
BAA4 7D          MODI2:  LD      A,L          ┐
BAA5 D3E8                OUT     (0E8H),A     │
BAA7 7C                  LD      A,H          │
BAA8 D3E9                OUT     (0E9H),A     │
BAAA D3EA                OUT     (0EAH),A     │
BAAC 00                  NOP                  │
BAAD 23                  INC     HL           │
BAAE DBE9                IN      A,(0E9H)     │
BAB0 12                  LD      (DE),A       │
BAB1 13                  INC     DE           ├・漢字ROMからデータを読んで表示
BAB2 DBE8                IN      A,(0E8H)     │
BAB4 12                  LD      (DE),A       │
BAB5 D3EB                OUT     (0EBH),A     │
BAB7 1B                  DEC     DE           │
BAB8 05                  DEC     B            │
BAB9 C8                  RET     Z            │
BABA 3E50                LD      A,80         │
BABC CDECB9              CALL    ADEAA        │
BABF 18E3                JR      MODI2        ┘

BAC1 01020402    KIGOU:  DB      1,2,4,2,5,2  ┐
BAC5 0502                                     │
BAC7 01030405    KUTEN:  DB      1,3,4,5,1,2  │
BACB 0102                                     │
BACD 01040504    EIMOJ:  DB      1,4,5,4,1,1  │
BAD1 0101                                     │
BAD3 01030602    SUJ08:  DB      1,3,6,2,3,1  │
BAD7 0301                                     │
BAD9 01030404    SUJD9:  DB      1,3,4,4,3,1  │
BADD 0301                                     ├・文字の種類別縮小データ
BADF 02040402    KAHIR:  DB      2,4,4,2,3,1  │
BAE3 0301                                     │
BAE5 01030404    HIGA1:  DB      1,3,4,4,3,1  │
BAE9 0301                                     │
BAEB 01020305    HIGA2:  DB      1,2,3,5,4,1  │
BAEF 0401                                     │
BAF1 01030303    KATA1:  DB      1,3,3,3,5,1  │
BAF5 0501                                     │
BAF7 01020403    KANJI:  DB      1,2,4,3,5,1  │
BAFB 0501                                     ┘

BAFD 7C          KIADR:  LD      A,H          ┐
BAFE E607                AND     00000111B    │
BB00 67                  LD      H,A          │
BB01 7D                  LD      A,L          │
BB02 E660                AND     01100000B    ├・漢字ROMアドレスを得る－①
BB04 0F                  RRCA                 │
BB05 0F                  RRCA                 │
BB06 1807                JR      GTKAD        ┘

BB08 7C          KAADR:  LD      A,H          ┐
BB09 E61F                AND     00011111B    │
BB0B 67                  LD      H,A          │
BB0C 7D                  LD      A,L          │
BB0D E660                AND     01100000B    │
BB0F B4          GTKAD:  OR      H            │
BB10 67                  LD      H,A          │
BB11 7D                  LD      A,L          │
BB12 07                  RLCA                 ├・漢字ROMアドレスを得る－②
BB13 07                  RLCA                 │
BB14 07                  RLCA                 │
BB15 07                  RLCA                 │
BB16 CB14                RL      H            │
BB18 E6F0                AND     11110000B    │
BB1A 6F                  LD      L,A          │
BB1B C9                  RET                  ┘
```

実行すると、画面には特徴を示すメッセージが表示されます。メッセージにも行末禁則処理*の例がありますが、次に示されるものに対しては自動改行で行頭に来ないようになっています。

＿　＿＿　　・・・・　スペース／ダブルスペース
。　、．？！　・・・・　句読点／エンドマーク
」　」」　@」　・・・・　」　』　）

文字データというのは、意外とメモリを消費しますから最初から省メモリ化を図る気持ちが大切です。そこで、本プログラムでは原則として非漢字1バイト、漢字2バイトで表すようにし、さらに通常の文字データをそのままサブルーチン的に利用できるようにしています。これは、スタックエリアがある限りいくらでもネスティングできます。

次に、ラベルの内容と文字列作成の秘訣を示します。

●ラベルの内容

@END　：文字列の終わりを示す。

@WT　：次の1バイト（1/0）で、1文字表示後のウェイト（オン／オフ）を示す。実際のウェイトルーチン（WAIT）は、内容に合わせ適宜変化させること。

@MAX　：次の1バイトで1行の最大コマ数（1文字＝2コマ）を示す。これを越えると行末禁則処理または自動改行となる。

@NEX　：次の2バイトで改行幅を示す。行間2ライン（通常）であれば、80*13となる。

@SUB　：次の2バイトでサブデータ・アドレスを示す。

@DJP　：次の2バイトでデータのジャンプ先（アドレス）を示す。

@HI　：ひらがな／漢字モード選択。

@KA　：カタカナ／漢字モード選択。

@EI　：漢字モードを英数字モードとする。英数字は引用符（'）で囲む。漢字モードには、@HI／@KAで戻る。

@GI　：漢字モードをギリシア文字モードとする。ギリシア文字は、JIS コードの下位1バイトで指定する。漢字モードには、@HI／@KAで戻る。

禁則処理：

禁則処理はラベル（＿　＿＿　、。．？！　」　」」　@」）で示された場合に限られる。記号としてJISコード番号で入力したものは、通常の改行処理がされる。

@RO　　：漢字モードをロシア文字モードとする。ロシア文字は、JISコードの下位1バイトで指定する。漢字モードには、@HI／@KAで戻る。

@NY　　：改行

_　　：スペース（1コマ）

_ _　　：ダブルスペース（1文字分）

、　　：、

。　　：。

.　　：.

?　　：?

!　　：!

」　　：」

」」　　：』

@」　　：）

-　　：ー（マイナス記号ではなく、カナ・ロックで¥キーを押したもの）

・　　：・

・・・　　：…

「　　：「

「「　　：『

@「　　：（

@KI1　　：次の1バイトを記号（2120H～217FH）の下位1バイトとして表示する。1回限り有効。

@KI2　　：次の1バイトを記号（2220H～222FH）の下位1バイトとして表示する。1回限り有効。

@SPK　　：次に来る漢字コード（2バイト）＋縮小法（6バイト）で、特別縮小の漢字を表示する。1回限り有効。縮小法は図3-5-6に示されるように、先頭の1ラインが前のラインと合成される。

(例) 1、3、3、4、4、1の場合

・6バイトの合計は必ず16にする
・0として、そのラインを無視することはできない
・表示されたラインの最終ラインは改行されない(次のラインと合成される)

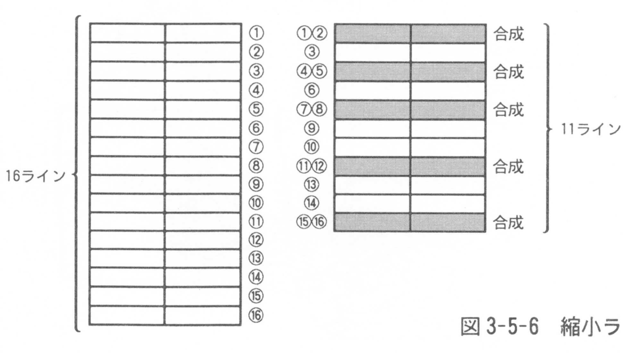


図3-5-6　縮小ラインの決め方

@GAI　：次に示されるアドレス（2バイト）にあるデータ（22バイト）
を、外字データとして表示する。1回限り有効。

ｱ〜ｳﾞ　：ひらがな／カタカナの文字にそのまま相当する。

C0〜C7　：文字のカラー番号。

●文字列作成の秘訣

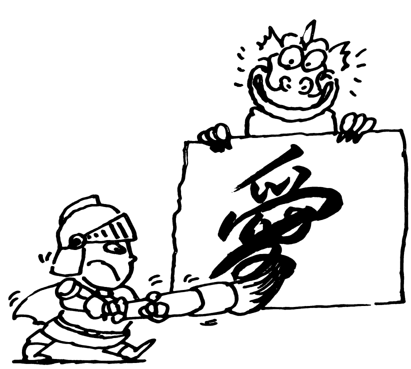
プログラム中の文章（KDATA）では、サンプルとして色々なモードを示しましたが、実際にはカナ文字と漢字コード（上位／下位の順）の入力がほとんどです。入力しながら漢字コードを捜すよりも、最初にすべての漢字コードを調べておくほうが作業効率はよくなります。

よく使用する文字列や特別縮小の漢字などはサブデータ化してメモリの有効利用を図ってください。ただし、サブデータ側でひらがな／カタカナのモードを切り換えたりカラーを変えると、メインの文章にはそのモードで戻りますから注意が必要です。また、サブデータのラベルはカナ表記にしておくとメインの文章がわかりやすくなります。

このプログラムを実行中は常にグラフィック画面を選択した状態ですから、プログラム本体および文字列データは、必ずC000H以前に置かれていなければなりません。特に、データ領域に関しては見逃しがちですから注意してください。文字列のDBとDWも間違えないように。

さて、このプログラムにはテクニック的に見るべき点があります。それは、サブデータを利用する際の再帰的（リカーシブル）用法です。これは自分で自分のいるプログラムをコールするという珍しいテクニックの1つです。

プログラムの「RECAL」の部分がそれに当り、サブデータを利用する際には文字列表示のメインルーチン「CKCOM」が、新たに「CKCOM」をコールします。もしもサブデータがネスティングされていると、「CKCOM」にコールされた「CKCOM」がさらに「CKCOM」をコールするという珍現象が起こるわけです。縮小テクニックの陰に隠れ、またプログラムとしても特にどうということのないサブルーチン・コールですが、こういった一人芝居的な用法はかなり高度なテクニックに属するものです。サブデータを使う際には、そんなこともぜひ思い出してください。

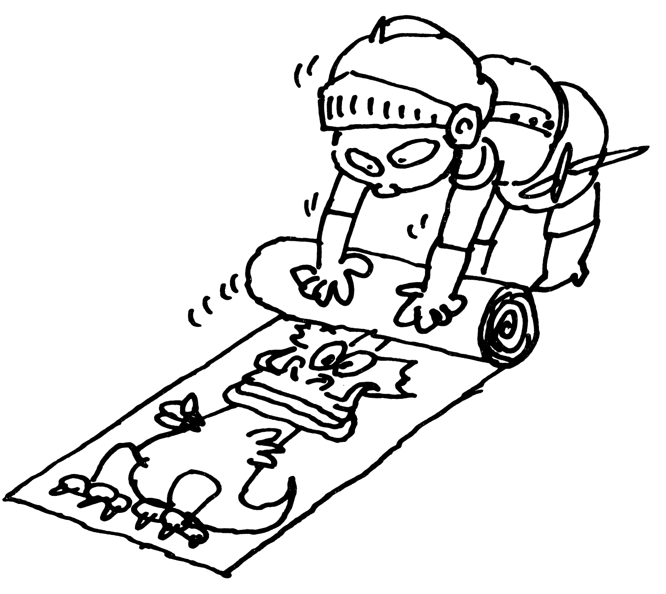


第4章　スクロール・テクニック

PC-8801シリーズは、8ビットのパソコンでありながらも、いわゆるゲーム専用機ではない本格パソコンとして商品化されています。上位機種であるPC-9801シリーズと変わらないキーボードを持ち、スプライト*やハードウェア・スクロールといったゲーム・マシンに必携の機能はありません。しかし、価格体系からいってユーザーのターゲットがゲーム中心の若者であることは事実でしょう。

ない機能はプログラム・テクニックでカバーする……当然のことですが、結果としてゲーム専用機のハードウェア機能を越えるテクニックを産み出したのですから、世の中なにが幸いするかわかりません。最近のゲームでは、多重重ね合わせ／多重スクロールといったゲーム専用機の上をいく技術も見られます。

もしも、メーカーにそこまでの先見の明があったとしたら……。これは、とてつもないゲーム専用機だったのかもしれません。



スプライト：

スプライトとは、背景との重ね合わせや移動時の消去などを意識せずにキャラクタを表示できる特殊なグラフィックス機能のこと。ハードウェア・スクロールとは、グラフィックVRAMのアドレスを変えることができる機能。画面全体がそれだけで変化するので、スクロールゲームには特に便利。

4-1 スクロールとキャラクタ

スクロールを辞書で調べると「巻物」という意味です。これが転じてスクロールゲームとなったわけですが、背景が一方的に動いていく初期の縦スクロールゲームは、まさに巻物というイメージにピッタリでした。

時は流れ、横スクロール、縦横スクロール、縦横斜めスクロール……と次第に巻物のイメージは薄れ、最近では多重スクロールという立体感あるものまで登場しています。これも技術の進歩があればこそなのですが、この技術を支えているのは拡張グラフィックス機能を含んだハードウェアの進化です。それだけに、対応機種がすべてPC-8801SR以降となるのも仕方のないことでしょう。

ところで、スクロール・ゲームといえば忘れてならないのが『マシン語ゲームプログラミング』掲載の『スカイ・ブルーザー』。画面構成こそグラフィックVRAMを背景とキャラクタとに分けたものですが、部分描き換え方式という手法は、いまでも重要なスクロール技法の1つとなっています。ハードウェア・スクロール機能のないPC-8801では、拡張グラフィックス(3面同時転送)による画面データ転送か、部分描き換えによる画面疑似移動によってのみスクロールは実現されるのです。

ただ、いずれの場合も問題は背景とキャラクタをどう区別するかです。『スカイ・ブルーザー』ではこれをプレーン分割によって解決していましたが、いまではフルカラーで重ね合わせ処理を行いながらスクロールさせるのは当然のこと。実際には、これこそがスクロール・テクニックの正体ともいえるくらいです。

こういったテクニックが成立するためには、まずスクロールの条件を確定する必要があります。実際のゲームであれば、敵の出現、歩ける／歩けないの区別、入口やトラップの区別、高さ／奥行きの情報……等、ゲームに合わせてマップデータの構造を細かく設定することから始めなければなりません。とりあえず、ここではマップパターン／スクロールの単位を横16ドット×縦8ドットとすることだけを条件にテクニックに迫ってみたいと思います。

この単位は、もちろんゲームによって自由に設定して構いませんし、これがベストというわけでもありません。しかし、最近のゲームの傾向と多重重ね合わせ等の高度なテクニックには、この程度の単位がちょうどいいのです。その理由は、スクロールの滑らかさと速度のバランスが適度に同居し、マップパターンと同一サイズということでプログラミングがしやすいからです。

では、最初に拡張グラフィックスの3面同時転送機能を使った場合のスクロールを考えてみましょう。図4-1-1は最も基本的なスクロール画面で、画面中央にいるキャラクタが下方向に歩いたときの処理状態を示したものです。疑似3D処理のアクションRPGなどでよく見かける光景です。

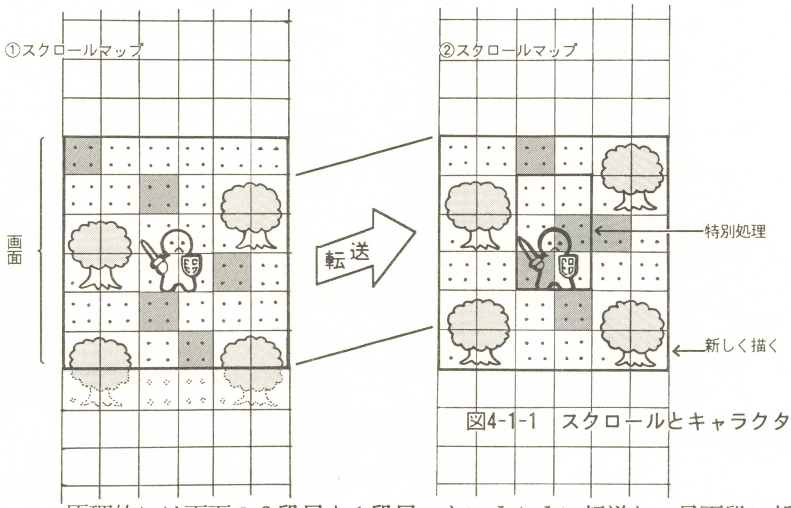


図4-1-1　スクロールとキャラクタの関係

原理的には画面の２段目を１段目へというふうに転送し、最下段に新しく現れるマップパターンを描けばいいのですが、キャラクタが関わる特別処理枠内は①のデータをそのまま転送するわけにはいきません。これに対する最も単純な解決策は、キャラクタ部分をいったん背景で消去し、画面転送により新しい背景が完成してから新たなキャラクタを表示するという方法です。

しかし、この方法はキャラクタ不在の時間が長く、キャラクタがチラつくという致命的な欠点があります。そこで、最初に特別処理部分をグラフィックVRAMの余り（FE80H ～）に作成しておき、横１ラインを次のように分割して画面転送するのです。

特別処理枠の左側：通常の画面転送
特別処理枠の部分：FE80H以降から転送
特別処理枠の右側：通常の画面転送

プログラム的にはレジスタのヤリクリが大変ですが、こうすることによりキャラクタの描き換え（消去／表示）とスクロールが同時に行われることになり、チラツキはまったくなくなります。もちろん、方向別に違ったプログラムを用意しなければなりませんが、基本的な考え方は同じです。画面転送プログラムを組む際には、方向（転送先）によってキチンとLDI命令とLDD命令を使い分けることも忘れないでください。

なお、画面がスクロールした場合、敵など主人公以外のキャラクタは相対的に反対方向へ移動したことになりますから、それらはすべてスクロールに合わせて座標を変化させなければなりません。これは『スカイ・ブルーザー』のように一方的な背景スクロールにはなかった意識の変化で、キー操作によってスクロールをコントロールする場合の基本的な注意事項です。

次に、もう１つのスクロール・テクニック、部分描き換えで図4-1-1と同じ内容を実現してみましょう。部分描き換えというのは、元の絵と違う部分（パターン）

だけを描き換えるということですから、図4-1-2に示される部分を新しく表示し直すわけです。

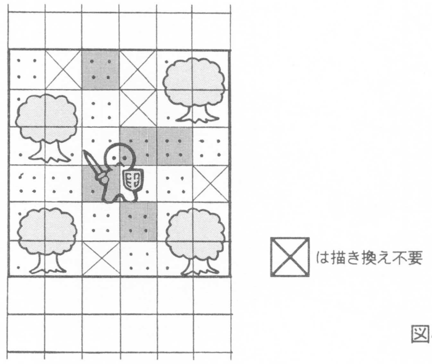


図4-1-2　部分描き換えによるスクロール

この例では5カ所が描き換え不要となりますが、マップによってあるいはスクロールする方向によってもその数は変わってきます。全面描き換えでは滑らかなスクロールが得られませんから、マップ作成の時点でできるだけ同じパターンが連続するよう配慮をしなければなりません。つまり、マップにある程度の制限が加えられるわけで、この点に関する限りは自由にパターンを組み合わせられる画面転送方式のほうが優れていると言えるでしょう。

また、上下にスクロールするときはパターンを上から順に（横列ごとに）、左右にスクロールするときは左から順に（縦列ごとに）描き換えるという配慮もスクロール時のブレを解消するテクニックの1つです。もちろん、描き換えるパターン数が増えれば、ブレは消えても新たに波打ち現象が発生しますから、スクロール方向に合わせて同一パターンを連続するという努力は常に欠かすことができません。

こうして見ると、部分描き換え方式にはメリットがなさそうですが、実際にはこちらのほうが発展性のあるスクロールなのです。というのは、スクロール処理とキャラクタ表示をパターン単位で同時に行うため、複数マップによる多重スクロール（複数の背景＋キャラクタの合成）が可能になるからです。

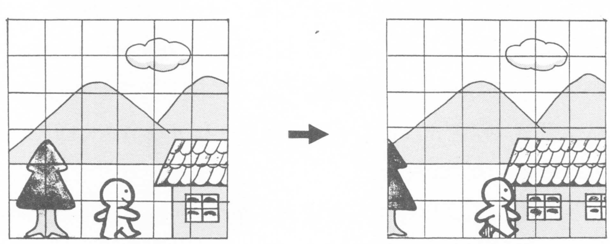


図4-1-3　多重スクロール

例えば、図4-1-3では山や雲の部分はそのままで、手前の木や家だけが横スクロールしていますが、画面転送ではこういった部分スクロールは不可能です。しかも、マップの枚数を追加してマップごとに移動頻度を変化させれば、さらに立体感のある画面を構成することができます（図4-1-4）。

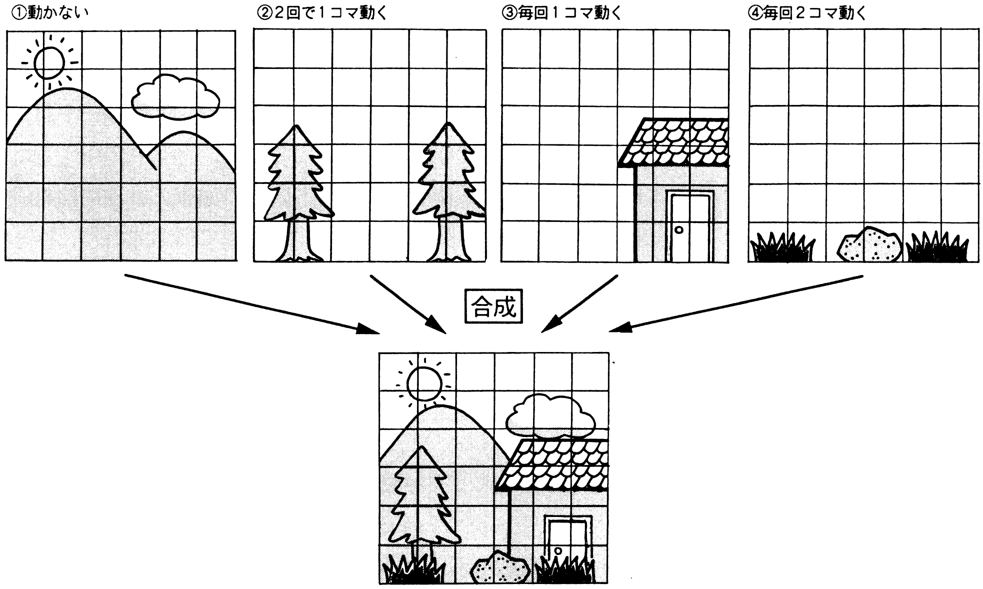


図4-1-4　マップの合成

背景の優先順位は①→④の順に高くなり、それにつれて1回のスクロールでの動きも大きくなります。これが多重スクロールによる奥行きの表現です。この図ではキャラクタを表示していませんが、キャラクタの扱いは従来のように単体での消去／表示という考え方ではなく、どちらかというとスプライトに近くなります。つまり、キャラクタも複数のパターンによって構成されたマップの一種と考え、座標やパターンの変化を部分描き換えの対象とするわけです。

当然、キャラクタをどのマップ上に重ね合わせるかという情報も必要です。常識的には2番目くらいの優先順位（③の上に重ね合わせる）となるはずですが、場合によってはキャラクタ間で変化をつけるのも味なものです。ただし、キャラクタどうしで重なった場合は、キャラクタ番号で優先順位をつけるのが普通です。このあたり、いかにもスプライト的な処理であり、ソフト・スプライトと言ってもいいかもしれません。

この節では、具体的なプログラム例は示しませんが、こういった高度なテクニックはアルゴリズムを正確に把握することが実践への第一歩です。まずは、最近のゲームを思い出しながらスクロールの基礎原理を理解してください。

4-2 スクロールと非スクロール

未確認飛行物体……。いわゆるUFOは、慣性を無視したジグザグ飛行や急発進／急停止を平気で行うそうです。地球上の乗り物であれば、方向を変換するには放物線を描くように反転するか、減速→停止→逆方向への加速という過程を経るはずです。つまり、そういった動きがわれわれ人間にとって自然であり、いきなり反転するという動きは不自然なのです。不自然が現実に起こるとすれば、それは衝突（事故）による慣性エネルギーの破壊でしかありません。

こう考えると、たとえゲームの世界であっても、キャラクタ（主人公）がキー操作により突然方向変更することは不自然であると言えます。とはいえ、ゲーム中にキャラクタが思い通りに動かないのでは、これまた不自然でイライラしてしまいます。結局、キー操作通りにキャラクタを動かすしかないのですが……。

スクロール・ゲームでは、主人公の動きは背景のスクロールを意味しますから、方向転換をすればいきなり背景が反対方向へ動くことになります。これが絶対的に悪いということではありませんが、減速～停止～加速に代わる緩衝剤として非スクロール・エリア（このエリア内はキャラクタが移動する）を設けてやると、スクロールの動きにゆとりができます（図4-2-1参照）。

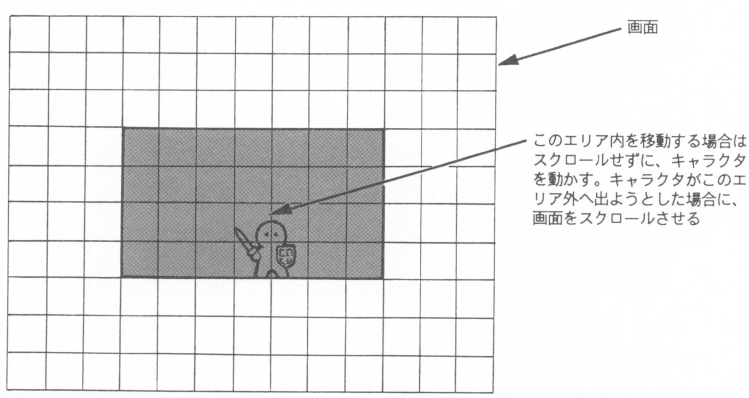


図4-2-1　非スクロール・エリアの例

これは、要するにキャラクタ（主人公）が画面中央に常駐しているのか、ある程度動き回るのかという違いなのですが、これによってマップが広く感じるという利点も生まれてきます。というのは、いずれにせよマップエンドになれば画面の端（通常は障害物でマップの周囲は閉鎖されている）までキャラクタが動くわけですが、その時に初めて画面中央から移動するよりも、普段から一定の範囲内を動いているほうがマップエンドの意識が少なくなるからです。ただし、非スクロール・エリアを広く取れば取るほど、画面の端から出現する敵との間合いが狭くなります。ゲームの内容と条件をよく考えて、そのことがネックにならないようにしなければなりません。

参考例として、図4-2-2を見てください。これは断面図タイプのスクロールですが、AからBへキャラクタ（主人公）がジャンプする際の動き（①→③）を示したものです。非スクロール・エリアがあるせいで、頂点までスクロールアップした後はキャラクタ自身が落下（③）しており、スクロールはありません。たとえ右側のタナがない場合でも、スクロールダウンするのは非スクロール・エリアから出ようとする時点ですから、視覚的にも目が慣れて見やすくなります。

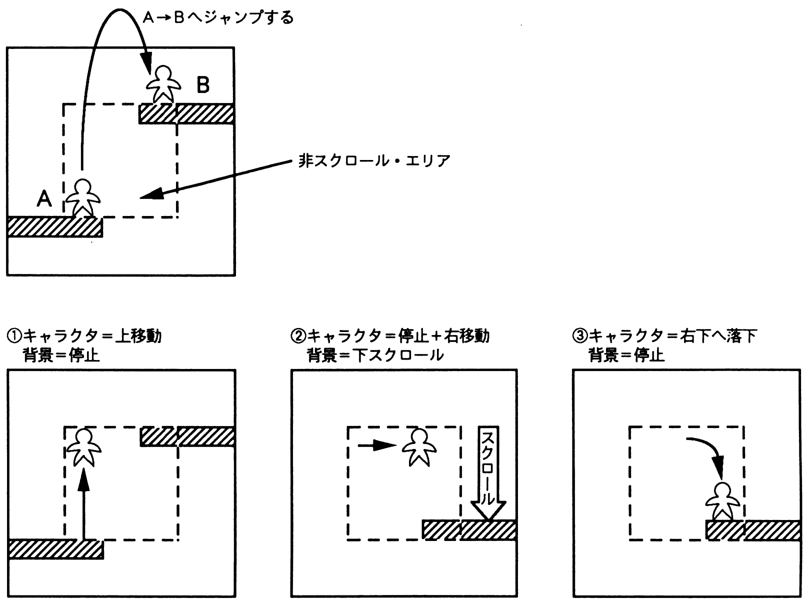


図4-2-2　スクロールと非スクロール

こういった動きの基本にプログラム・テクニック（「4-1 スクロールとキャラクタ」の内容）が加わって、初めてスクロール・テクニックとなるわけですが、これだけではまだプログラムを組むことはできません。スクロール・ゲームにとって背景をスクロールさせる（またはキャラクタを動かす）ことは、プログラム以前の必要最低条件に過ぎないのです。重要なのは、例えば図4-2-2でいうならば、どうしてAからBへジャンプした際にタナから落下しないのかということなのです。

これまではマップを単なるパターン番号の集まりとして認識してきましたが、実際には「4-1 スクロールとキャラクタ」でも触れたように、敵の出現、歩ける／歩けない、入口、各種トラップ、高さ（多重重ね合わせの場合）、奥行き（多重スクロールの場合）……等、ゲームに合わせて様々な情報が必要です。そして、これらの情報にこそ本当のテクニックが潜んでいるのです。

キャラクタや画面をコントロールするのはテクニックの氷山の一角、いわばスクロールの当然の知識です。水面下には画面に現れない多くの情報が隠されており、ここにゲームの実体があるのです。

4-3 マップの秘密

アクションタイプのRPGでは、あまり数字遊びという感覚はありませんが、思考型の場合は所持金やH.P./M.P.といったものをヤリクリするのに相当頭を使います。ある意味では、それがゲームそのものであり、イベントや謎などは「数字遊び」をゲームとして成立させるための手段のような気さえします。

最近流行の戦略シミュレーション・ゲームも、シナリオはどうあれ結局は数字をもてあそぶゲームですし、「アチラを立てればコチラが立たず」という数字のお遊びは、それだけでパズルとしての面白さがあるのかもしれません。

実は、似たようなことはプログラムを組む前の段階から行われています。限られたメモリの中に、いかにして多くの情報を組み入れるか……。これは、プログラミング前の頭のトレーニング、すなわちパズルでもあります。例えば、『スカイ・ブルーザー』では背景パターンを0〜127で表し、ビット7を敵出現のフラグにしていました。当然、敵の種類を示すマップが別に必要となります（図4-3-1）。

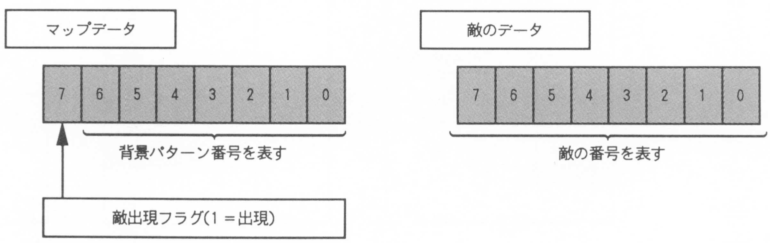


図4-3-1 『スカイ・ブルーザー』のマップデータ構造

この場合、一方通行型のスクロールということで敵マップから不要部分（敵の出現しない部分）をカットしていましたが、自在にスクロール方向を選択できるタイプでは、敵マップもベタで持たないと、背景マップと一致させるのが大変です。その結果、マップに必要なメモリ数は単純に背景マップの2倍となります。この2枚のマップは、多重スクロール用のマップのように独立したものではなく、ビット不足を補うためのものですから、マップを2バイト単位（背景用1バイト＋敵用1バイト）とすれば、形の上では1つのマップにすることもできます。

このような方式で次々とその他の情報を追加していくと、マップデータはどんどん膨らんでしまい、結果としてオンメモリでのマップは小さくなる一方です。そこで、若干の制限はありますが、情報が1バイトに納まるようデータに工夫を凝らしてみましょう。

●制限事項

敵の種類　　　　　：16種（0〜15）以下に限定
敵の出現する背景：パターン番号0〜7に限定

この制限がゲームに及ぼす影響は、ほとんどゼロに近いものです。というの

は、オンメモリでは敵の種類は16もあれば十分ですし、出現位置もこれだけの選択余地があればまず支障はないからです。それどころか、この程度の制限が問題になるようであれば、それはマッピングの努力不足というべきもの。マップ・エディタさえ制限に合わせて作成しておけば、実際には悩むこともないはずです。

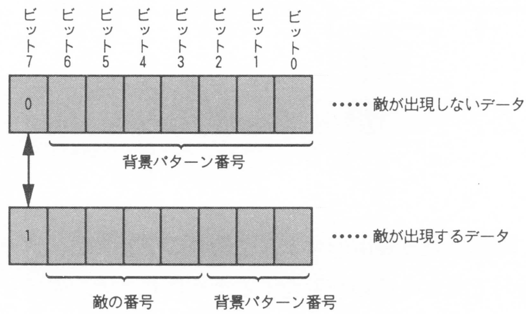


図 4-3-2

データの構造は、敵の出現フラグ（ビット7）が立っていなければ従来通り、立っている場合にはビット3〜6で敵の番号を表すというものです。貴重なメモリですから、ゲームデザインからの協力も必要なのです。

一方、歩ける／歩けない、入口やトラップ……など、場所に関する情報を追加したい場合は、パターン番号で区別します。当然、プログラム側で区別しやすいようパターンを種類別にまとめておくことが大切です。ただし、特定の入口や個々に内容が違うトラップの場合は、さらに座標で区別することになります。

また、ふかん図のような疑似３Ｄ画面では、よく逆重ね合わせのテクニックが使われます。例えば、人が柵の手前にいる時は人を優先し、柵の奥にいる時は柵を優先して表示するという具合いです（図 4-3-3）。

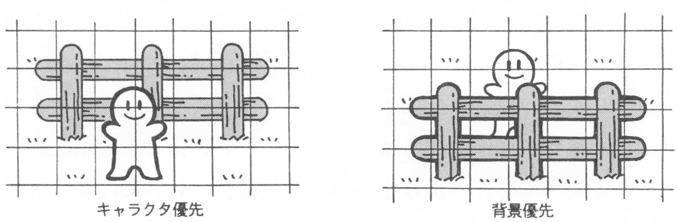


図4-3-3　逆重ね合わせ

こういったことを実現するには、背景パターン／キャラクタに高さの情報を持たせ、それぞれの高さを比較して優先順位をつければいいのです（図 4-3-4）。

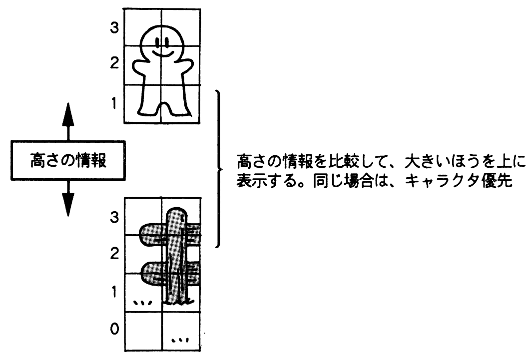


図4-3-4　高さの情報

これも、理想は高さ専用のマップを持つことですが、メモリ節約のため同様にパターン番号で高さの情報を管理するようにします。先ほどの歩ける／歩けないの情報と試しに組み合わせてみましょう。

パターン番号	歩行	高さ	グラフィック内容
0〜19	可	0	地上
20〜39	不可	1	棚の下部や木の根元など
40〜49	可	2	棚の中間部や木の幹など
50〜69	可	3	棚の上部や木の上部など
70〜79	不可	2	家の壁や屋根などの一部
80〜89	可	0	トラップ（落とし穴）など
⋮	⋮	⋮	⋮

地上のパターン（0〜7）には敵出現の情報が入りますから、同系統のパターンが片寄らないようにする工夫も必要です。また、地形からどうしてもそれが無理な場合は、特殊処理としてパターン番号で敵を出現させることもあります。

ここに示した内容は単なるサンプルに過ぎませんが、このようにパターン番号に情報を定義すると、ほとんどの情報は１枚のマップに収めることができ、オンメモリで広い世界を表現することができるようになります。

キャラクタ移動時には、足元のマップを見て移動可能かどうかを判定し、移動後の表示は高さの情報を参照して重ね合わせの順位を決めます。この時、逆重ね合わせ（キャラクタの上に背景の一部が表示される）の可能性のある背景パターンは、パターンを分割するデータが必要です。つまり、キャラクタが上位にくる場合は通常の重ね合わせ処理（背景＋キャラクタ）ですが、背景の一部が上位にくる場合は、背景パターンを地面（キャラクタより下位に表示される）と柵（キャラクタより上位に表示される）とに分ける必要があるということです。

これをプログラムで実現すると、次のような処理手順となります。なお、ここでの手順はPC-8801SR以降を対象としており、拡張グラフィックスのALU機能を利用することを前提としています。したがって、キャラクタの透明（くり抜き）データは、透明部分のビット＝0、キャラクタ部分のビット＝1となっています。一方、背景パターンの分割データは、図4-3-5で図示されているように、下になる部分のビット＝1、上になる部分のビット＝0とします。

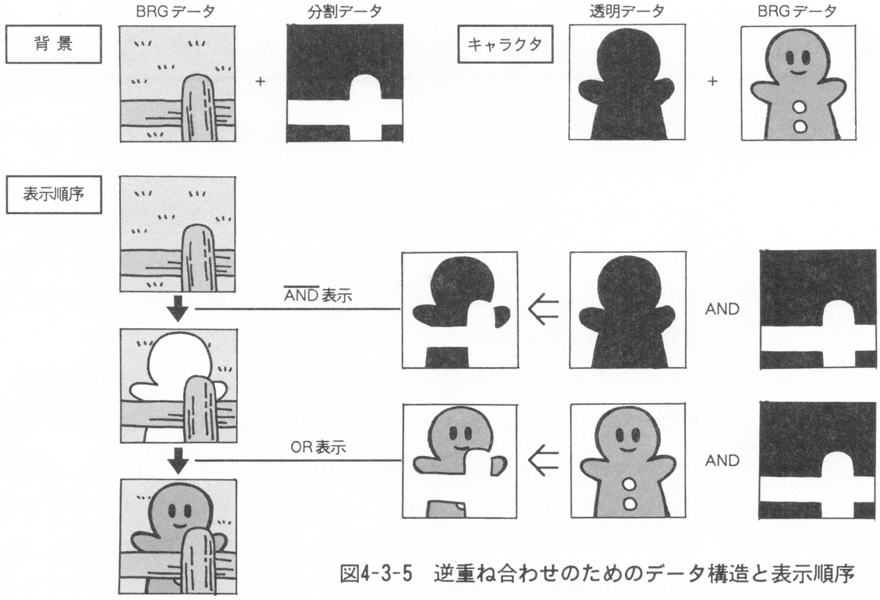


図4-3-5　逆重ね合わせのためのデータ構造と表示順序

①背景パターンを表示
②ALUのビットリセット(BRG面)モードにする
③透明データを分割データとANDを取ってからG.VRAMにストア
④ALUのビットセット(B面のみ)モードにする
⑤キャラクタ・データ(B面)を分割データとANDを取ってからG.VRAMにストア
⑥ALUのビットセット(R面のみ)モードにする
⑦キャラクタ・データ(R面)を分割データとANDを取ってからG.VRAMにストア
⑧ALUのビットセット(G面のみ)モードにする
⑨キャラクタ・データ(G面)を分割データとANDを取ってからG.VRAMにストア

この一連の処理は、グラフィックVRAMの余り（FE80H以降）で行ってから実際の表示先へ転送したほうがいいことは、すでに第2章「2-5　グラフィックVRAMの余り」で述べた通りです。スクロールというのは、最も速度を要求されるテクニックですから、すべての面で高速化を追求しなければならないのです。

なお、歩ける／歩けないのチェックについては、方向によって条件が変化する場合もあります。特に、断面図タイプのスクロールでは、図4-2-2のようにタナがズレているとは限らず、真上にタナがあるケースも少なくありません。このような場合、上方向には進めても下方向には進めない（落ちない）ようにプログラムを組まなければなりません。

また、断面図タイプで上方向へ進む時には、足元ではなく頭の部分で進める／進めないのチェックをします。その際、見かけは同じようでも天井の場合もありますから、両者はキチンと区別をつけておかなければなりません。いずれにしても、この段階でパターン番号に意味を持たせて整理するということは、プログラム前のパズルであり、最初の難所といっても過言ではないでしょう。

4-4 多重スクロールのサンプル

スクロールに関する限り、これまでの説明で少なくとも「わからないからできない」という段階は過ぎたことでしょう。拡張グラフィックスの機能もすべて知っているし、重ね合わせの方法もマスターしてきました。あとは、作りたいゲームに合わせてプログラムを組むばかり。……と、すんなりコトが運べば苦労はしませんね。「わからないからできない」の次には「わかっていてもできない」という難関が控えています。

本来、アルゴリズムがわかればプログラムは組めるはずですが、フルカラースクロールの場合は、そのアルゴリズムを実現するためにこれまでに紹介してきたいくつものテクニックを複合して応用しなければなりません。実際問題として、これはスクロールの原理を理解する以上に難しいことです。すべてのスクロールをサンプル・プログラムとして掲載できればいいのですが、ページに限りのある本ではそれも無理な話です。

そこで、数あるスクロールの中から最も新しいテクニックである多重スクロールを取り上げ、これを徹底的に解析してこの難関を乗り越えてもらうことにしました。多重スクロールというのは、部分描き換えによる単純スクロールを発展させたものですが、断面図タイプの画面に奥行き感を出したり、立体駐車場のような階層を表現できるので、今後の応用が期待されているテクニックのひとつです。

また、スクロール処理と切り離せないのがキャラクタとの合成です。「4-1 スクロールとキャラクタ」でソフト・スプライトという言葉が出てきましたが、キャラクタをスクロール画面と無関係に動かすことは、プログラム的にはスクロール以上に面倒なことです。しかし、ソフト・スプライトを省いてしまってはゲームになりませんね。基本的な考え方は「4-1 スクロールとキャラクタ」に示したとおりですが、実際には複雑な処理を経て重ね合わせ処理が実現されています。

では、まずはサンプル・プログラムを実行して、こういった2つのテクニックを自分の目で確認してください。

```
LINE(0,0)-(639,183),1,BF
LINE(61,6)-(578,137),7,BF
CLEAR,&HB5FF:CLS:A=&HB700:CALL A
```

LIST 4.1

```
                           ORG     0B600H

B600 000004B6   ITABL:     DW      0,VRTCT          ・割込みテーブル

B604 F3         VRTCT:     DI                      |
B605 F5                    PUSH    AF              |
B606 3A1DB6                LD      A,(VWAIT+1)     |
B609 3C                    INC     A               |
B60A 321DB6                LD      (VWAIT+1),A     |・割込み処理ルーチン
```

```
B60D 3E02                 LD      A,2
B60F D3E4                 OUT     (0E4H),A
B611 F1                   POP     AF
B612 FB                   EI
B613 C9                   RET

B614 F3          VWCHK:   DI
B615 DB09                 IN      A,(9)
B617 07                   RLCA
B618 07                   RLCA
B619 30F9                 JR      NC,VWCHK
B61B FB                   EI                         ・ウェイト（スペースが押されている間はループ）
B61C 3E00        VWAIT:   LD      A,0
B61E FE05                 CP      5
B620 38FA                 JR      C,VWAIT
B622 AF                   XOR     A
B623 321DB6               LD      (VWAIT+1),A
B626 C9                   RET

                          ORG     0B700H

C288=            DITOP:   EQU     0C000H+80*8+8      ・マップ表示位置
0020=            DMAXX:   EQU     32                 ・表示パターン数（X）
0010=            DMAXY:   EQU     16                 ・表示パターン数（Y）

0002=            FPAT1:   EQU     2                  ・この番号以降ベタ・パターン（背景②）
0002=            FPAT2:   EQU     2                  ・この番号以降ベタ・パターン（背景③）

F980=            SPGVA:   EQU     0F980H             ・ソフト・スプライト用G.VRAM
0014=            SPVAL:   EQU     20                 ・総スプライト数

FE80=            URAGV:   EQU     0FE80H             ・パターン合成エリア（G.VRAM）
FE90=            GPT00:   EQU     URAGV+16           ・ベタ・パターンデータ（背景①／番号0）
FEA0=            GPT01:   EQU     GPT00+16           ・ベタ・パターンデータ（背景①／番号1）
FEB0=            GPT11:   EQU     GPT01+16           ・ベタ・パターンデータ（背景②／番号2）
FEC0=            GPT21:   EQU     GPT11+16           ・ベタ・パターンデータ（背景③／番号2）
FED0=            GPT22:   EQU     GPT21+16           ・ベタ・パターンデータ（背景③／番号3）

B700 F3                   DI                         ・割込み禁止
B701 ED7393B7             LD      (SPPOP+1),SP       ・SPの退避と設定
B705 3100B7               LD      SP,0B700H
B708 2190FE               LD      HL,0FE90H
B70B 1191FE               LD      DE,0FE91H
B70E 3687                 LD      (HL),'■'
B710 014F00               LD      BC,79              ・テキスト画面を部分マスク
B713 EDB0                 LDIR
B715 3E08                 LD      A,00001000B
B717 13                   INC     DE
B718 12                   LD      (DE),A
B719 D35C                 OUT     (5CH),A
B71B 21E8BC               LD      HL,BKD00
B71E 1190FE               LD      DE,GPT00
B721 012000               LD      BC,16*2
B724 EDB0                 LDIR
B726 CD97B7               CALL    GVTEN              ・データをG.VRAMに転送
B729 11C0FE               LD      DE,GPT21
B72C CD97B7               CALL    GVTEN
B72F 11D0FE               LD      DE,GPT22
B732 CD97B7               CALL    GVTEN
B735 D35F                 OUT     (5FH),A
B737 DB32                 IN      A,(32H)
B739 CBF7                 SET     6,A                ・拡張アクセスモード
B73B D332                 OUT     (32H),A
B73D CD98BD               CALL    ALMAP              ・初期画面用描き換え情報マップ作成
B740 CDB3B7               CALL    DIMAP              ・初期画面表示
B743 3EB6                 LD      A,0B6H
B745 ED47                 LD      I,A
B747 3E02                 LD      A,2                ・割込み関係の設定
B749 D3E6                 OUT     (0E6H),A
B74B D3E4                 OUT     (0E4H),A
```

```
B74D FB                           EI                                   ・背景②③とピーヨを動かして表示
B74E CDF2BD       MLOOP:          CALL     MVMP1
B751 CD29BE                       CALL     MVMP2
B754 CDDABE                       CALL     QSSET
B757 CD6BBF                       CALL     PIYOM
B75A CD8AB9                       CALL     MAKSG
B75D CDB3B7                       CALL     DIMAP
B760 CD14B6                       CALL     VWCHK               ・ウェイトのチェック
B763 CDD3BD                       CALL     NOMP1               ・背景③とピーヨを動かして表示
B766 CD29BE                       CALL     MVMP2
B769 CDDABE                       CALL     QSSET
B76C CD6BBF                       CALL     PIYOM
B76F CD8AB9                       CALL     MAKSG
B772 CDB3B7                       CALL     DIMAP
B775 CD14B6                       CALL     VWCHK               ・ウェイトのチェック
B778 DB09                         IN       A,(9)               ・[ESC] が押されていなければMLOOPへ
B77A 07                           RLCA
B77B 38D1                         JR       C,MLOOP
B77D F3                           DI                           ・割込み関係を元に戻す
B77E 3EF3                         LD       A,0F3H
B780 ED47                         LD       I,A
B782 3AC3E6                       LD       A,(0E6C3H)
B785 D3E4                         OUT      (0E4H),A
B787 3A0EEF                       LD       A,(0EF0EH)
B78A D3E6                         OUT      (0E6H),A
B78C DB32                         IN       A,(32H)             ・独立アクセスモード
B78E CBB7                         RES      6,A
B790 D332                         OUT      (32H),A
B792 310000       SPPOP:          LD       SP,0                ・SPを元に戻す
B795 FB                           EI                           ・割込み許可
B796 C9                           RET                          ・BASICへ戻る

B797 D35C         GVTEN:          OUT      (5CH),A             ・データをG.VRAMへ転送
B799 CDA3B7                       CALL     GTENS
B79C D35D                         OUT      (5DH),A
B79E CDA3B7                       CALL     GTENS
B7A1 D35E                         OUT      (5EH),A
B7A3 D5           GTENS:          PUSH     DE
B7A4 011000                       LD       BC.16
B7A7 EDB0                         LDIR
B7A9 D1                           POP      DE
B7AA C9                           RET

B7AB 000068BC     MA1TB:          DW       0,BKT11             ・背景②重ね合わせパターン・テーブル
B7AF 0000A8BC     MA2TB:          DW       0,BKT21             ・背景③重ね合わせパターン・テーブル

B7B3 FD213AC0     DIMAP:          LD       IY,CKMAP            ・IY＝描き換え情報マップアドレス
B7B7 213AC2                       LD       HL,MAP00            ・各マップアドレス初期化
B7BA 225FB8                       LD       (DMAP0+1),HL
B7BD 213AC4                       LD       HL,MAP01
B7C0 22CDB8                       LD       (DMAP1+1),HL
B7C3 214AC6                       LD       HL,MAP02
B7C6 22DBB8                       LD       (DMAP2+1),HL
B7C9 2180F9                       LD       HL,SPGVA
B7CC 227DB9                       LD       (SPRIT+1),HL
B7CF D9                           EXX                          ・裏レジスタ内容
B7D0 2188C2                       LD       HL,DITOP              HL＝表示アドレス
B7D3 118002                       LD       DE,80*8               DE＝次パターン表示への加算バイト数
B7D6 0E20                         LD       C,DMAXX               C＝表示パターン数 (X)
B7D8 0610         DMPL0:          LD       B,DMAXY               B＝表示パターン数 (Y)
B7DA E5                           PUSH     HL
B7DB D9           DMPL1:          EXX                          ・描き換え情報マップの内容により分岐
B7DC FD7E00                       LD       A,(IY+0)
B7DF 0F                           RRCA
B7E0 D2A7B8                       JP       NC,MPSAM
B7E3 0F                           RRCA
B7E4 DA7CB9                       JP       C,SPRIT
B7E7 0F                           RRCA
B7E8 DA7CB8                       JP       C,FULL2
B7EB 0F                           RRCA
```

```
B7EC DA06B8              JP      C,CMAP1
B7EF 0F                  RRCA
B7F0 DA88B8              JP      C,FULL1
B7F3 0F                  RRCA
B7F4 D294B8              JP      NC,FULL0
B7F7 CD5EB8              CALL    DMAP0
B7FA CDCCB8              CALL    DMAP1
B7FD 1180FE              LD      DE,URAGV          ・背景①（フル表示）＋背景②（重ね合わせ表示）
B800 CD2FB9              CALL    PUTPT               ＋背景③（表示不要）
B803 C3B5B8              JP      MPSA1
B806 0F         CMAP1:   RRCA                      ・情報により分岐
B807 302A                JR      NC,CMAP2
B809 2ACDB8              LD      HL,(DMAP1+1)
B80C 7E                  LD      A,(HL)
B80D 23                  INC     HL
B80E 22CDB8              LD      (DMAP1+1),HL
B811 D602                SUB     FPAT1
B813 87                  ADD     A,A
B814 87                  ADD     A,A
B815 87                  ADD     A,A
B816 87                  ADD     A,A
B817 5F                  LD      E,A               ・背景①（表示不要）＋背景②（フル表示）
B818 1600                LD      D,0                 ＋背景③（重ね合わせ表示）
B81A 21B0FE              LD      HL,GPT11
B81D CD70B8              CALL    DMASB
B820 CDDAB8              CALL    DMAP2
B823 1180FE              LD      DE,URAGV
B826 CD2FB9              CALL    PUTPT
B829 2A5FB8              LD      HL,(DMAP0+1)
B82C 23                  INC     HL
B82D 225FB8              LD      (DMAP0+1),HL
B830 C3BCB8              JP      MPNEX
B833 0F         CMAP2:   RRCA                      ・情報により分岐
B834 3012                JR      NC,CMAP3
B836 CD5EB8              CALL    DMAP0
B839 CDCCB8              CALL    DMAP1
B83C CDDAB8              CALL    DMAP2             ・背景①（フル表示）＋背景②（重ね合わせ表示）
B83F 1180FE              LD      DE,URAGV            ＋背景③（重ね合わせ表示）
B842 CD2FB9              CALL    PUTPT
B845 C3BCB8              JP      MPNEX
B848 CD5EB8     CMAP3:   CALL    DMAP0
B84B CDDAB8              CALL    DMAP2
B84E 1180FE              LD      DE,URAGV
B851 CD2FB9              CALL    PUTPT             ・背景①（フル表示）＋背景②（表示不要）＋背景③
B854 2ACDB8              LD      HL,(DMAP1+1)        （重ね合わせ表示）
B857 23                  INC     HL
B858 22CDB8              LD      (DMAP1+1),HL
B85B C3BCB8              JP      MPNEX

B85E 213AC2     DMAP0:   LD      HL,MAP00
B861 7E                  LD      A,(HL)
B862 23                  INC     HL
B863 225FB8              LD      (DMAP0+1),HL
B866 87                  ADD     A,A
B867 87                  ADD     A,A
B868 87                  ADD     A,A
B869 87                  ADD     A,A
B86A 5F                  LD      E,A
B86B 1600                LD      D,0               ・背景①の表示
B86D 2190FE              LD      HL,GPT00
B870 19         DMASB:   ADD     HL,DE
B871 3E90                LD      A,10010000B
B873 D335                OUT     (35H),A
B875 CD0BB9              CALL    DURA2
B878 AF                  XOR     A
B879 D335                OUT     (35H),A
B87B C9                  RET

B87C 2ADBB8     FULL2:   LD      HL,(DMAP2+1)
B87F 7E                  LD      A,(HL)
```

```
B880 D602              SUB    FPAT2            ・背景③だけのパターン
B882 21C0FE            LD     HL,GPT21
B885 C39BB8            JP     FUL12
B888 2ACDB8   FULL1:   LD     HL,(DMAP1+1)     ・背景②だけのパターン
B88B 7E                LD     A,(HL)
B88C D602              SUB    FPAT1
B88E 21B0FE            LD     HL,GPT11
B891 C39BB8            JP     FUL12
B894 2A5FB8   FULL0:   LD     HL,(DMAP0+1)     ・背景①だけのパターン
B897 7E                LD     A,(HL)
B898 2190FE            LD     HL,GPT00
B89B 87       FUL12:   ADD    A,A
B89C 87                ADD    A,A
B89D 87                ADD    A,A
B89E 87                ADD    A,A
B89F 5F                LD     E,A
B8A0 1600              LD     D,0
B8A2 19                ADD    HL,DE
B8A3 EB                EX     DE,HL
B8A4 CD2FB9            CALL   PUTPT
B8A7 2A5FB8   MPSAM:   LD     HL,(DMAP0+1)     ・マップアドレスを変更
B8AA 23                INC    HL
B8AB 225FB8            LD     (DMAP0+1),HL
B8AE 2ACDB8            LD     HL,(DMAP1+1)
B8B1 23                INC    HL
B8B2 22CDB8            LD     (DMAP1+1),HL
B8B5 2ADBB8   MPSA1:   LD     HL,(DMAP2+1)
B8B8 23                INC    HL
B8B9 22DBB8            LD     (DMAP2+1),HL
B8BC FD23     MPNEX:   INC    IY               ・各レジスタを次パターン用に変更し必要回数だけ
B8BE D9                EXX                       ループ
B8BF 19                ADD    HL,DE
B8C0 05                DEC    B
B8C1 C2DBB7            JP     NZ,DMPL1
B8C4 E1                POP    HL
B8C5 23                INC    HL
B8C6 23                INC    HL
B8C7 0D                DEC    C
B8C8 C2D8B7            JP     NZ,DMPL0
B8CB C9                RET

B8CC 213AC4   DMAP1:   LD     HL,MAP01         ・背景②の表示
B8CF 7E                LD     A,(HL)
B8D0 23                INC    HL
B8D1 22CDB8            LD     (DMAP1+1),HL
B8D4 21ABB7            LD     HL,MA1TB
B8D7 C3E5B8            JP     DICHR

B8DA 214AC6   DMAP2:   LD     HL,MAP02         ・背景③の表示
B8DD 7E                LD     A,(HL)
B8DE 23                INC    HL
B8DF 22DBB8            LD     (DMAP2+1),HL
B8E2 21AFB7            LD     HL,MA2TB

B8E5 87       DICHR:   ADD    A,A              ・マップパターンをFE80H (G.VRAM) からに作成
B8E6 5F                LD     E,A
B8E7 1600              LD     D,0
B8E9 19                ADD    HL,DE
B8EA 7E                LD     A,(HL)
B8EB 23                INC    HL
B8EC 66                LD     H,(HL)
B8ED 6F                LD     L,A
B8EE 3E80              LD     A,10000000B
B8F0 D335              OUT    (35H),A
B8F2 AF                XOR    A
B8F3 CD09B9            CALL   DIURA
B8F6 3E67              LD     A,01100111B
B8F8 CD09B9            CALL   DIURA
B8FB 3E57              LD     A,01010111B
B8FD CD09B9            CALL   DIURA
```

```
B900 3E37                  LD      A,00110111B
B902 CD09B9                CALL    DIURA
B905 AF                    XOR     A
B906 D335                  OUT     (35H),A
B908 C9                    RET
B909 D334       DIURA:     OUT     (34H),A
B90B 1180FE     DURA2:     LD      DE,URAGV
B90E EDA0 *10              LDI     16
B92E C9                    RET

B92F 3E90       PUTPT:     LD      A,10010000B
B931 D335                  OUT     (35H),A
B933 D9                    EXX
B934 E5                    PUSH    HL
B935 D9                    EXX
B936 E1                    POP     HL
B937 014F00                LD      BC,79
B93A 1A                    LD      A,(DE)
B93B 77                    LD      (HL),A
B93C 13                    INC     DE
B93D 23                    INC     HL
B93E 1A                    LD      A,(DE)
B93F 77                    LD      (HL),A
B940 13                    INC     DE
B941 09                    ADD     HL,BC
B942 1A                    LD      A,(DE)
B943 77                    LD      (HL),A
B944 13                    INC     DE
B945 23                    INC     HL
B946 1A                    LD      A,(DE)
B947 77                    LD      (HL),A
B948 13                    INC     DE
B949 09                    ADD     HL,BC
B94A 1A                    LD      A,(DE)
B94B 77                    LD      (HL),A
B94C 13                    INC     DE
B94D 23                    INC     HL
B94E 1A                    LD      A,(DE)
B94F 77                    LD      (HL),A
B950 13                    INC     DE
B951 09                    ADD     HL,BC
B952 1A                    LD      A,(DE)
B953 77                    LD      (HL),A
B954 13                    INC     DE
B955 23                    INC     HL
B956 1A                    LD      A,(DE)
B957 77                    LD      (HL),A
B958 13                    INC     DE
B959 09                    ADD     HL,BC
B95A 1A                    LD      A,(DE)
B95B 77                    LD      (HL),A
B95C 13                    INC     DE
B95D 23                    INC     HL
B95E 1A                    LD      A,(DE)
B95F 77                    LD      (HL),A
B960 13                    INC     DE
B961 09                    ADD     HL,BC
B962 1A                    LD      A,(DE)
B963 77                    LD      (HL),A
B964 13                    INC     DE
B965 23                    INC     HL
B966 1A                    LD      A,(DE)
B967 77                    LD      (HL),A
B968 13                    INC     DE
B969 09                    ADD     HL,BC
B96A 1A                    LD      A,(DE)
B96B 77                    LD      (HL),A
B96C 13                    INC     DE
B96D 23                    INC     HL
B96E 1A                    LD      A,(DE)
```

・作成済みのパターンを裏レジスタのHLで指定される位置に転送

```
B96F 77                 LD      (HL),A
B970 13                 INC     DE
B971 09                 ADD     HL,BC
B972 1A                 LD      A,(DE)
B973 77                 LD      (HL),A
B974 13                 INC     DE
B975 23                 INC     HL
B976 1A                 LD      A,(DE)
B977 77                 LD      (HL),A
B978 AF                 XOR     A
B979 D335               OUT     (35H),A
B97B C9                 RET

B97C 1180F9    SPRIT:   LD      DE,SPGVA
B97F CD2FB9             CALL    PUTPT
B982 13                 INC     DE                   ・スプライト・パターンの転送
B983 ED537DB9           LD      (SPRIT+1),DE
B987 C3A7B8             JP      MPSAM

B98A CD04BF    MAKSG:   CALL    MLANK                ・各スプライトの表示順位を決める(結果＝SLANK～)
B98D 2180F9             LD      HL,SPGVA
B990 22C5BA             LD      (SGADR+1),HL
B993 3E01               LD      A,1
B995 21C5B9    MSPLP:   LD      HL,MSPON+1
B998 3601               LD      (HL),1
B99A 21D9BE             LD      HL,SLANK+SPVAL-1     ・順位通りにスプライト表示パターンを作成する
B99D 0614               LD      B,SPVAL
B99F BE        MSPL1:   CP      (HL)
B9A0 2804               JR      Z,MSPG1
B9A2 2B                 DEC     HL
B9A3 10FA               DJNZ    MSPL1
B9A5 C9                 RET
B9A6 F5        MSPG1:   PUSH    AF
B9A7 E5                 PUSH    HL
B9A8 C5                 PUSH    BC
B9A9 CDD3B9             CALL    MGVDT                ・1ケ目のスプライト表示パターンを作成
B9AC C1                 POP     BC
B9AD E1                 POP     HL
B9AE F1                 POP     AF
B9AF C3BFB9             JP      MSPG3
B9B2 2B        MSPG2:   DEC     HL
B9B3 BE                 CP      (HL)
B9B4 2009               JR      NZ,MSPG3
B9B6 F5                 PUSH    AF
B9B7 E5                 PUSH    HL
B9B8 C5                 PUSH    BC
B9B9 CD19BA             CALL    MGV1W                ・同一順位のスプライトがあればパターンを作成
B9BC C1                 POP     BC
B9BD E1                 POP     HL
B9BE F1                 POP     AF
B9BF 10F1      MSPG3:   DJNZ    MSPG2
B9C1 CD2BBA             CALL    MGVD2
B9C4 C601      MSPON:   ADD     A,1
B9C6 2AC5BA             LD      HL,(SGADR+1)
B9C9 111000             LD      DE,16                ・背景③のパターンを重ね合わせ、スプライト番号／
B9CC 19                 ADD     HL,DE                  表示アドレスを進める
B9CD 22C5BA             LD      (SGADR+1),HL
B9D0 C395B9             JP      MSPLP

B9D3 78        MGVDT:   LD      A,B
B9D4 07                 RLCA                         ・スプライト・パターンのアドレスを求めるため
B9D5 320CBA             LD      (MGPAT+1),A
B9D8 80                 ADD     A,B
B9D9 4F                 LD      C,A
B9DA 0600               LD      B,0
B9DC 2162BE             LD      HL,SPWOK
B9DF 09                 ADD     HL,BC
B9E0 2B                 DEC     HL
B9E1 4E                 LD      C,(HL)
B9E2 2B                 DEC     HL                   ・背景③のマップアドレスを求めるため
```

```
B9E3 6E                 LD      L,(HL)
B9E4 2600               LD      H,0
B9E6 29                 ADD     HL,HL
B9E7 29                 ADD     HL,HL
B9E8 29                 ADD     HL,HL
B9E9 29                 ADD     HL,HL
B9EA 09                 ADD     HL,BC
B9EB 226CBA             LD      (MP2BC+1),HL
B9EE 44                 LD      B,H
B9EF 4D                 LD      C,L
B9F0 113AC0             LD      DE,CKMAP            ・描き換え情報マップのアドレスをセット
B9F3 19                 ADD     HL,DE
B9F4 221EBA             LD      (MGVD1+1),HL
B9F7 222CBA             LD      (MGVD2+1),HL
B9FA CBC6               SET     0,(HL)              ・描き換え情報マップに描き換え有りのフラグを立てる
B9FC CBCE               SET     1,(HL)              ・描き換え情報マップにスプライト有りのフラグを立てる
B9FE CB56               BIT     2,(HL)              ・背景③がベタ・パターンならリターン
BA00 C0                 RET     NZ
BA01 CB66               BIT     4,(HL)              ・背景②がﾍﾞﾀ･ﾊﾟﾀｰﾝでないなら背景①のﾊﾟﾀｰﾝを作成
BA03 CC37BA             CALL    Z,SFMP0
BA06 CB6E               BIT     5,(HL)              ・背景②が空間でなければ背景②のパターンを作成
BA08 C445BA             CALL    NZ,SSMP1
BA0B 010000    MGPAT:   LD      BC,0
BA0E 219CBE    MGPA2:   LD      HL,SPPAT-2
BA11 09                 ADD     HL,BC
BA12 CDA4BA             CALL    SPCH2               ・指定のスプライト・パターンを作成
BA15 AF                 XOR     A
BA16 D335               OUT     (35H),A
BA18 C9                 RET

BA19 21C5B9    MGV1W:   LD      HL,MSPON+1          ・次の表示順位への加算数を＋１する
BA1C 34                 INC     (HL)
BA1D 210000    MGVD1:   LD      HL,0
BA20 CB56               BIT     2,(HL)              ・背景③がベタ・パターンならリターン
BA22 C0                 RET     NZ
BA23 78                 LD      A,B
BA24 07                 RLCA
BA25 4F                 LD      C,A                 ・スプライト・パターンを作成
BA26 0600               LD      B,0
BA28 C30EBA             JP      MGPA2

BA2B 210000    MGVD2:   LD      HL,0
BA2E CB5E               BIT     3,(HL)              ・背景③が空間（パターンなし）ならリターン
BA30 C8                 RET     Z
BA31 F5                 PUSH    AF
BA32 CD68BA             CALL    SSMP2               ・背景③のパターン作成
BA35 F1                 POP     AF
BA36 C9                 RET

BA37 C5        SFMP0:   PUSH    BC
BA38 D9                 EXX
BA39 C1                 POP     BC
BA3A 213AC2             LD      HL,MAP00            ・背景①のパターンを指定スプライトエリアに作成
BA3D 09                 ADD     HL,BC
BA3E 7E                 LD      A,(HL)
BA3F 2190FE             LD      HL,GPT00
BA42 C37BBA             JP      SMFDI

BA45 CB66      SSMP1:   BIT     4,(HL)              ・背景②が重ね合わせパターンならSPMP1へ
BA47 CA5ABA             JP      Z,SPMP1
BA4A C5        SFMP1:   PUSH    BC
BA4B D9                 EXX
BA4C C1                 POP     BC
BA4D 213AC4             LD      HL,MAP01
BA50 09                 ADD     HL,BC               ・ベタ・パターン（背景②）を指定スプライトエリア
BA51 7E                 LD      A,(HL)                に作成
BA52 D602               SUB     FPAT1
BA54 21B0FE             LD      HL,GPT11
BA57 C37BBA             JP      SMFDI
```

```
BA5A C5         SPMP1:  PUSH  BC
BA5B D9                 EXX
BA5C C1                 POP   BC
BA5D 213AC4             LD    HL,MAP01          ・重ね合わせパターン（背景②）を指定スプライトエ
BA60 09                 ADD   HL,BC               リアに作成
BA61 7E                 LD    A,(HL)
BA62 21ABB7             LD    HL,MA1TB
BA65 C397BA             JP    SPMPS

BA68 CB56       SSMP2:  BIT   2,(HL)
BA6A D9                 EXX                     ・背景③が重ね合わせパターンならSPMP2へ
BA6B 010000     MP2BC:  LD    BC,0
BA6E CA8FBA             JP    Z,SPMP2
BA71 214AC6     SFMP2:  LD    HL,MAP02
BA74 09                 ADD   HL,BC
BA75 7E                 LD    A,(HL)
BA76 D602               SUB   FPAT2
BA78 21C0FE             LD    HL,GPT21
BA7B 87         SMFDI:  ADD   A,A
BA7C 87                 ADD   A,A
BA7D 87                 ADD   A,A
BA7E 87                 ADD   A,A
BA7F 5F                 LD    E,A               ・ベタ・パターン（背景③）を指定スプライトエリア
BA80 1600               LD    D,0                 に作成
BA82 19                 ADD   HL,DE
BA83 3E90               LD    A,10010000B
BA85 D335               OUT   (35H),A
BA87 CDC4BA             CALL  SGADR
BA8A AF                 XOR   A
BA8B D335               OUT   (35H),A
BA8D D9                 EXX
BA8E C9                 RET

BA8F 214AC6     SPMP2:  LD    HL,MAP02
BA92 09                 ADD   HL,BC
BA93 7E                 LD    A,(HL)
BA94 21AFB7             LD    HL,MA2TB
BA97 CD9FBA     SPMPS:  CALL  SPCHR             ・重ね合わせパターン（背景③）を指定スプライト
BA9A AF                 XOR   A                   エリアに作成
BA9B D335               OUT   (35H),A
BA9D D9                 EXX
BA9E C9                 RET

BA9F 87         SPCHR:  ADD   A,A
BAA0 5F                 LD    E,A
BAA1 1600               LD    D,0
BAA3 19                 ADD   HL,DE
BAA4 7E         SPCH2:  LD    A,(HL)
BAA5 23                 INC   HL
BAA6 66                 LD    H,(HL)
BAA7 6F                 LD    L,A
BAA8 3E80               LD    A,10000000B
BAAA D335               OUT   (35H),A
BAAC AF                 XOR   A
BAAD D334               OUT   (34H),A           ・指定のスプライトエリア（SGADR＋1）に
BAAF CDC4BA             CALL  SGADR               パターンを重ね合わせる
BAB2 3E67               LD    A,01100111B
BAB4 D334               OUT   (34H),A
BAB6 CDC4BA             CALL  SGADR
BAB9 3E57               LD    A,01010111B
BABB D334               OUT   (34H),A
BABD CDC4BA             CALL  SGADR
BAC0 3E37               LD    A,00110111B
BAC2 D334               OUT   (34H),A
BAC4 110000     SGADR:  LD    DE,0
BAC7 EDA0 *10           LDI   16
BAE7 C9                 RET
                                                ・スプライト（ピーヨ）データ・0
BAE8 003F01FF   SPAT0:  DB    0,3FH,1,0FFH,7,0FFH,0FH,0FFH
BAEC 07FF0FFF
```

```
BAF0 1FFF3FFF         DB    1FH,0FFH,3FH,0FFH,7FH,0FFH,7FH,0FFH
BAF4 7FFF7FFF
BAF8 0000003F         DB    0,0,0,3FH,1,0FFH,7,0FFH
BAFC 01FF07FF
BB00 0FF01FE0         DB    0FH,0F0H,1FH,0E0H,3FH,0E0H,3FH,0F0H
BB04 3FE03FF0
BB08 0000000B         DB    0,0,0,0BH,0,5FH,1,70H
BB0C 005F0170
BB10 02EF05DF         DB    2,0EFH,5,0DFH,0BH,0DCH,5,0EFH
BB14 0BDC05EF
BB18 0000003F         DB    0,0,0,3FH,1,0FFH,7,0F0H
BB1C 01FF07F0
BB20 0FEF1FDF         DB    0FH,0EFH,1FH,0DFH,3FH,0DCH,3FH,0EFH
BB24 3FDC3FEF
                                          ・スプライト（ピーヨ）データ・1
BB28 F800FF00 SPAT1:  DB    0F8H,0,0FFH,0,0FFH,0C0H,0FFH,0E0H
BB2C FFC0FFE0
BB30 FFF0FFF8         DB    0FFH,0F0H,0FFH,0F8H,0FFH,0F8H,0FFH,0F8H
BB34 FFF8FFF8
BB38 0000F800         DB    0,0,0F8H,0,0FFH,0,0FFH,0C0H
BB3C FF00FFC0
BB40 F0E06070         DB    0F0H,0E0H,60H,70H,60H,70H,0F0H,0F0H
BB44 6070F0F0
BB48 00006000         DB    0,0,60H,0,0FCH,0,0F0H,80H
BB4C FC00F080
BB50 6F409FA0         DB    6FH,40H,09FH,0A0H,093H,0A0H,6FH,40H
BB54 93A06F40
BB58 0000F800         DB    0,0,0F8H,0,0FFH,0,0F0H,0C0H
BB5C FF00F0C0
BB60 6F609FB0         DB    6FH,60H,09FH,0B0H,093H,0B0H,6FH,70H
BB64 93B06F70
                                          ・スプライト（ピーヨ）データ・2
BB68 7FFF7FFF SPAT2:  DB    7FH,0FFH,7FH,0FFH,3FH,0FFH,5FH,0FFH
BB6C 3FFF5FFF
BB70 FFFF7FFF         DB    0FFH,0FFH,7FH,0FFH,3FH,0FFH,0FH,0FFH
BB74 3FFF0FFF
BB78 3FFF3FFF         DB    3FH,0FFH,3FH,0FFH,1EH,4CH,0FH,27H
BB7C 1E4C0F27
BB80 470731FF         DB    47H,7,31H,0FFH,0FH,0FFH,0,0
BB84 0FFF0000
BB88 0BF005FF         DB    0BH,0F0H,5,0FFH,4,0ECH,3,77H
BB8C 04EC0377
BB90 0157107F         DB    1,57H,10H,7FH,5,55H,0,0
BB94 05550000
BB98 3FF03FFF         DB    3FH,0F0H,3FH,0FFH,1EH,0ECH,0FH,77H
BB9C 1EEC0F77
BBA0 435730FF         DB    43H,57H,30H,0FFH,0FH,0FFH,0,0
BBA4 0FFF0000
                                          ・スプライト（ピーヨ）データ・3
BBA8 FFF8FFFC SPAT3:  DB    0FFH,0F8H,0FFH,0FCH,0FFH,0FEH,0FFH,0FFH
BBAC FFFEFFFF
BBB0 FFFFFF3E         DB    0FFH,0FFH,0FFH,3EH,0F8H,0,80H,0
BBB4 F8008000
BBB8 FFF0FFF8         DB    0FFH,0F0H,0FFH,0F8H,0F9H,0D4H,7,0AAH
BBBC F9D407AA
BBC0 FF00F800         DB    0FFH,0,0F8H,0,80H,0,0,0
BBC4 80000000
BBC8 F0E0FFD8         DB    0F0H,0E0H,0FFH,0D8H,0F9H,09CH,6,3EH
BBCC F99C063E
BBD0 FE2A5000         DB    0FEH,2AH,50H,0,0,0,0,0
BBD4 00000000
BBD8 F0F0FFF8         DB    0F0H,0F0H,0FFH,0F8H,0F9H,0DCH,7,0BEH
BBDC F9DC07BE
BBE0 FF2AF800         DB    0FFH,2AH,0F8H,0,80H,0,0,0
BBE4 80000000
                                          ・スプライト（ピーヨ）データ・4
BBE8 7FFF7FFF SPAT4:  DB    7FH,0FFH,7FH,0FFH,7FH,0FFH,0FFH,0FFH
BBEC 7FFFFFFF
BBF0 7FFF3FFF         DB    7FH,0FFH,3FH,0FFH,7,0FFH,0,54H
BBF4 07FF0054
```

```
BBF8 3FFF3FFF            DB      3FH,0FFH,3FH,0FFH,1FH,6CH,5FH,77H
BBFC 1F6C5F77
BC00 3FF30769            DB      3FH,0F3H,7,69H,0,0,0,0
BC04 00000000
BC08 0BF005FF            DB      0BH,0F0H,5,0FFH,5,2CH,1,57H
BC0C 052C0157
BC10 017B007D            DB      1,7BH,0,7DH,0,54H,0,0
BC14 00540000
BC18 3FF03FFF            DB      3FH,0F0H,3FH,0FFH,1FH,6CH,47H,77H
BC1C 1F6C4777
BC20 3B7B077D            DB      3BH,7BH,7,7DH,0,54H,0,0
BC24 00540000
                                          ・スプライト (ピーヨ) データ・5
BC28 FFF8FFF8  SPAT5:    DB      0FFH,0F8H,0FFH,0F8H,0FFH,0F8H,0FFH,0FCH
BC2C FFF8FFFC
BC30 FFFCFFFE            DB      0FFH,0FCH,0FFH,0FEH,0F8H,0FEH,0,54H
BC34 F8FE0054
BC38 FFF0FFF0            DB      0FFH,0F0H,0FFH,0F0H,0F9H,0F0H,7,0F8H
BC3C F9F007F8
BC40 FF68F854            DB      0FFH,68H,0F8H,54H,0,0,0,0
BC44 00000000
BC48 F0E0FFC0            DB      0F0H,0E0H,0FFH,0C0H,0F9H,0A0H,7,0B0H
BC4C F9A007B0
BC50 FF38507C            DB      0FFH,38H,50H,7CH,0,54H,0,0
BC54 00540000
BC58 F0F0FFF0            DB      0F0H,0F0H,0FFH,0F0H,0F9H,0F0H,7,0B8H
BC5C F9F007B8
BC60 FF78F87C            DB      0FFH,78H,0F8H,7CH,0,54H,0,0
BC64 00540000
                                          ・背景②重ね合わせ・パターンデータ(パターン番号=1)
BC68 07E007E0  BKT11:    DB      7,0E0H,7,0E0H,7,0E0H,7,0E0H
BC6C 07E007E0
BC70 07E007E0            DB      7,0E0H,7,0E0H,7,0E0H,7,0E0H
BC74 07E007E0
BC78 02000200            DB      2,0,2,0,2,0,2,0
BC7C 02000200
BC80 02000200            DB      2,0,2,0,2,0,2,0
BC84 02000200
BC88 034003C0            DB      3,40H,3,0C0H,3,40H,3,0C0H
BC8C 034003C0
BC90 034003C0            DB      3,40H,3,0C0H,3,40H,3,0C0H
BC94 034003C0
BC98 03800300            DB      3,80H,3,0,3,80H,3,0
BC9C 03800300
BCA0 03800300            DB      3,80H,3,0,3,80H,3,0
BCA4 03800300
                                          ・背景③重ね合わせ・パターンデータ(パターン番号=1)
BCA8 00000180  BKT21:    DB      0,0,1,80H,3,0C0H,7,0E0H
BCAC 03C007E0
BCB0 0FF01FF8            DB      0FH,0F0H,1FH,0F8H,3FH,0FCH,7FH,0FEH
BCB4 3FFC7FFE
BCB8 00000000            DB      0,0,0,0,1,80H,3,0C0H
BCBC 018003C0
BCC0 07E00FF0            DB      7,0E0H,0FH,0F0H,1FH,0F8H,3FH,0FCH
BCC4 1FF83FFC
BCC8 00000000            DB      0,0,0,0,0,0,2,0
BCCC 00000200
BCD0 02000A00            DB      2,0,0AH,0,0AH,0,2AH,0
BCD4 0A002A00
BCD8 00000000            DB      0,0,0,0,1,80H,3,40H
BCDC 01800340
BCE0 07A00F50            DB      7,0A0H,0FH,50H,1FH,0A8H,3FH,54H
BCE4 1FA83F54
                                          ・背景①ベタ・パターンデータ (パターン番号=0)
BCE8 FFFFF7F7  BKD00:    DB      0FFH,0FFH,0F7H,0F7H,0DDH,0DDH,33H,33H
BCEC DDDD3333
BCF0 AAAA4444            DB      0AAH,0AAH,44H,44H,10H,10H,0,0
BCF4 10100000
                                          ・背景①ベタ・パターンデータ (パターン番号=1)
BCF8 00001010  BKD01:    DB      0,0,10H,10H,44H,44H,0AAH,0AAH
```

```
BCFC 4444AAAA
BD00 3333DDDD                DB     33H,33H,0DDH,0DDH,0F7H,0F7H,0FFH,0FFH
BD04 F7F7FFFF
                                         ・背景②ベタ・パターンデータ（パターン番号＝2）
BD08 6A205020    BKD12:      DB     6AH,20H,50H,20H,68H,20H,50H,0
BD0C 68205000
BD10 6C005400                DB     6CH,0,54H,0,6CH,0,40H,0
BD14 6C004000
BD18 7FDC7FDA                DB     7FH,0DCH,7FH,0DAH,7FH,0C4H,55H,52H
BD1C 7FC45552
BD20 7BF47BFA                DB     7BH,0F4H,7BH,0FAH,7BH,0E4H,55H,52H
BD24 7BE45552
BD28 7F8A7504                DB     7FH,8AH,75H,4,6AH,09AH,50H,0
BD2C 6A9A5000
BD30 6A8A7204                DB     6AH,8AH,72H,4,78H,09AH,50H,0
BD34 789A5000
                                         ・背景③ベタ・パターンデータ（パターン番号＝2）
BD38 6FFF6FFC    BKD22:      DB     6FH,0FFH,6FH,0FCH,6FH,0FFH,6FH,0FCH
BD3C 6FFF6FFC
BD40 6FFF6FFC                DB     6FH,0FFH,6FH,0FCH,6FH,0FFH,6FH,0FFH
BD44 6FFF6FFF
BD48 02AB02AB                DB     2,0ABH,2,0ABH,2,0ABH,2,0ABH
BD4C 02AB02AB
BD50 02AB02AB                DB     2,0ABH,2,0ABH,2,0ABH,0,0
BD54 02AB0000
BD58 5BBF5BBF                DB     5BH,0BFH,5BH,0BFH,5BH,0BFH,5BH,0BFH
BD5C 5BBF5BBF
BD60 5BBF5BBF                DB     5BH,0BFH,5BH,0BFH,5BH,0BFH,11H,15H
BD64 5BBF1115
                                         ・背景③ベタ・パターンデータ（パターン番号＝3）
BD68 3FF6FFF6    BKD23:      DB     3FH,0F6H,0FFH,0F6H,3FH,0F6H,0FFH,0F6H
BD6C 3FF6FFF6
BD70 3FF6FFF6                DB     3FH,0F6H,0FFH,0F6H,3FH,0F6H,0FFH,0F6H
BD74 3FF6FFF6
BD78 D540D540                DB     0D5H,40H,0D5H,40H,0D5H,40H,0D5H,40H
BD7C D540D540
BD80 D540D540                DB     0D5H,40H,0D5H,40H,0D5H,40H,0,0
BD84 D5400000
BD88 FDDAFDDA                DB     0FDH,0DAH,0FDH,0DAH,0FDH,0DAH,0FDH,0DAH
BD8C FDDAFDDA
BD90 FDDAFDDA                DB     0FDH,0DAH,0FDH,0DAH,0FDH,0DAH,0A8H,88H
BD94 FDDAA888

BD98 FD213AC0    ALMAP:      LD     IY,CKMAP
BD9C 213AC4                  LD     HL,MAP01
BD9F 114AC6                  LD     DE,MAP02
BDA2 010002                  LD     BC,DMAXX*DMAXY
BDA5 FD360001    ALMAL:      LD     (IY+0),1
BDA9 7E                      LD     A,(HL)
BDAA B7                      OR     A
BDAB 280C                    JR     Z,ALMP1
BDAD FDCB00EE                SET    5,(IY+0)
BDB1 FE02                    CP     FPAT1
BDB3 3804                    JR     C,ALMP1
BDB5 FDCB00E6                SET    4,(IY+0)
BDB9 1A          ALMP1:      LD     A,(DE)
BDBA B7                      OR     A             ・初期画面作成用に描き換え情報マップを作成
BDBB 280C                    JR     Z,ALMP2
BDBD FDCB00DE                SET    3,(IY+0)
BDC1 FE02                    CP     FPAT2
BDC3 3804                    JR     C,ALMP2
BDC5 FDCB00D6                SET    2,(IY+0)
BDC9 FD23        ALMP2:      INC    IY
BDCB 23                      INC    HL
BDCC 13                      INC    DE
BDCD 0B                      DEC    BC
BDCE 78                      LD     A,B
BDCF B1                      OR     C
BDD0 20D3                    JR     NZ,ALMAL
BDD2 C9                      RET
```

```
BDD3 213AC0    NOMP1:  LD     HL,CKMAP
BDD6 113AC4            LD     DE,MAP01
BDD9 010002            LD     BC,DMAXX*DMAXY
BDDC 3600      NOM1L:  LD     (HL),0
BDDE 1A                LD     A,(DE)
BDDF B7                OR     A
BDE0 2808              JR     Z,NOM11
BDE2 CBEE              SET    5,(HL)
BDE4 FE02              CP     FPAT1          ・背景②非スクロール時の描き換え情報マップを作成
BDE6 3802              JR     C,NOM11
BDE8 CBE6              SET    4,(HL)
BDEA 23        NOM11:  INC    HL
BDEB 13                INC    DE
BDEC 0B                DEC    BC
BDED 78                LD     A,B
BDEE B1                OR     C
BDEF 20EB              JR     NZ,NOM1L
BDF1 C9                RET

BDF2 213AC0    MVMP1:  LD     HL,CKMAP
BDF5 D9                EXX
BDF6 213AC4            LD     HL,MAP01
BDF9 113AC6            LD     DE,MAP1S
BDFC 011000            LD     BC,DMAXY
BDFF EDB0              LDIR
BE01 213AC4            LD     HL,MAP01
BE04 114AC4            LD     DE,MAP01+DMAXY
BE07 010002            LD     BC,DMAXX*DMAXY
BE0A 1A        MV1LP:  LD     A,(DE)
BE0B BE                CP     (HL)
BE0C 77                LD     (HL),A
BE0D D9                EXX
BE0E 3600              LD     (HL),0
BE10 2802              JR     Z,MVM11
BE12 CBC6              SET    0,(HL)         ・背景②スクロール時の描き換え情報マップを作成
BE14 B7        MVM11:  OR     A
BE15 2808              JR     Z,MVM12
BE17 CBEE              SET    5,(HL)
BE19 FE02              CP     FPAT1
BE1B 3802              JR     C,MVM12
BE1D CBE6              SET    4,(HL)
BE1F 23        MVM12:  INC    HL
BE20 D9                EXX
BE21 23                INC    HL
BE22 13                INC    DE
BE23 0B                DEC    BC
BE24 78                LD     A,B
BE25 B1                OR     C
BE26 20E2              JR     NZ,MV1LP
BE28 C9                RET

BE29 213AC0    MVMP2:  LD     HL,CKMAP
BE2C D9                EXX
BE2D 214AC6            LD     HL,MAP02
BE30 114AC8            LD     DE,MAP2S
BE33 011000            LD     BC,DMAXY
BE36 EDB0              LDIR
BE38 214AC6            LD     HL,MAP02
BE3B 115AC6            LD     DE,MAP02+DMAXY
BE3E 010002            LD     BC,DMAXX*DMAXY
BE41 1A        MV2LP:  LD     A,(DE)
BE42 FE02              CP     FPAT2
BE44 3806              JR     C,MVM20
BE46 D9                EXX
BE47 CB86              RES    0,(HL)
BE49 CBD6              SET    2,(HL)
BE4B D9                EXX
BE4C BE        MVM20:  CP     (HL)           ・背景③スクロールの描き換え情報マップを作成
BE4D 77                LD     (HL),A
BE4E D9                EXX
```

```
BE4F 2802                  JR      Z,MVM21
BE51 CBC6                  SET     0,(HL)
BE53 B7           MVM21:   OR      A
BE54 2802                  JR      Z,MVM22
BE56 CBDE                  SET     3,(HL)
BE58 23           MVM22:   INC     HL
BE59 D9                    EXX
BE5A 23                    INC     HL
BE5B 13                    INC     DE
BE5C 0B                    DEC     BC
BE5D 78                    LD      A,B
BE5E B1                    OR      C
BE5F 20E0                  JR      NZ,MV2LP
BE61 C9                    RET

BE62 00 *003C     SPWOK:   DUP     0,3*SPVAL      ・各スプライト（フラグ，X，Y）
BE9E 00 *0028     SPPAT:   DUP     0,2*SPVAL      ・各スプライト（パターンアドレス）
BEC6 00 *0014     SLANK:   DUP     0,SPVAL        ・各スプライト表示順位

BEDA 2162BE       QSSET:   LD      HL,SPWOK
BEDD 1600                  LD      D,0
BEDF 0614                  LD      B,SPVAL
BEE1 CB3E         QSTSL:   SRL     (HL)
BEE3 23                    INC     HL
BEE4 3805                  JR      C,QSSL1
BEE6 23                    INC     HL
BEE7 23                    INC     HL
BEE8 10F7                  DJNZ    QSTSL
BEEA C9                    RET
BEEB 5E           QSSL1:   LD      E,(HL)
BEEC 23                    INC     HL
BEED 7E                    LD      A,(HL)
BEEE 23                    INC     HL
BEEF D5                    PUSH    DE
BEF0 D9                    EXX                    ・使用中のスプライトの座標にあたる描き換え情報
BEF1 E1                    POP     HL              マップを作成
BEF2 29                    ADD     HL,HL
BEF3 29                    ADD     HL,HL
BEF4 29                    ADD     HL,HL
BEF5 29                    ADD     HL,HL
BEF6 113AC0                LD      DE,CKMAP
BEF9 19                    ADD     HL,DE
BEFA 5F                    LD      E,A
BEFB 1600                  LD      D,0
BEFD 19                    ADD     HL,DE
BEFE CBC6                  SET     0,(HL)
BF00 D9                    EXX
BF01 10DE                  DJNZ    QSTSL
BF03 C9                    RET

BF04 2162BE       MLANK:   LD      HL,SPWOK
BF07 11C6BE                LD      DE,SLANK
BF0A 0614                  LD      B,SPVAL
BF0C 7E           MLANL:   LD      A,(HL)
BF0D B7                    OR      A
BF0E 23                    INC     HL
BF0F C419BF                CALL    NZ,CHLCK       ・各スプライトに表示順位をつける
BF12 12                    LD      (DE),A
BF13 23                    INC     HL
BF14 23                    INC     HL
BF15 13                    INC     DE
BF16 10F4                  DJNZ    MLANL
BF18 C9                    RET

BF19 E5           CHLCK:   PUSH    HL
BF1A D5                    PUSH    DE
BF1B C5                    PUSH    BC
BF1C 56                    LD      D,(HL)
BF1D 23                    INC     HL
BF1E 5E                    LD      E,(HL)
```

```
BF1F 2162BE            LD      HL,SPWOK
BF22 010114            LD      BC,SPVAL*100H+1
BF25 CB46      CHLCL:  BIT     0,(HL)
BF27 23                INC     HL
BF28 280F              JR      Z,CHLC2
BF2A 7A                LD      A,D
BF2B 96                SUB     (HL)
BF2C 380B              JR      C,CHLC2
BF2E 2008              JR      NZ,CHLC1
BF30 23                INC     HL               ・スプライトの表示順位を各座標を調べて求める
BF31 7B                LD      A,E
BF32 96                SUB     (HL)
BF33 3805              JR      C,CHLC3
BF35 2803              JR      Z,CHLC3
BF37 2B                DEC     HL
BF38 0C        CHLC1:  INC     C
BF39 23        CHLC2:  INC     HL
BF3A 23        CHLC3:  INC     HL
BF3B 10E8              DJNZ    CHLCL
BF3D 79                LD      A,C
BF3E C1                POP     BC
BF3F D1                POP     DE
BF40 E1                POP     HL
BF41 C9                RET

BF42 02030100  PIYO0:  DB      2,3,1,0,0
BF46 00
BF47 0A060100  PIYO1:  DB      10,6,1,0,1
BF4B 01
BF4C 0F0B0100  PIYO2:  DB      15,11,1,0,0      ・ピーヨ・0～4（X, Y, カウンタ, 方向, ピーヨ番号）
BF50 00
BF51 160C0100  PIYO3:  DB      22,12,1,0,0
BF55 00
BF56 080E0100  PIYO4:  DB      8,14,1,0,0
BF5A 00

BF5B E8BA28BB  CHR00:  DW      SPAT0,SPAT1,SPAT2,SPAT3  ・ピーヨ番号＝0に用いるパターン
BF5F 68BBA8BB
BF63 E8BA28BB  CHR01:  DW      SPAT0,SPAT1,SPAT4,SPAT5  ・ピーヨ番号＝1に用いるパターン
BF67 E8BB28BC

BF6B 2162BE    PIYOM:  LD      HL,SPWOK
BF6E D9                EXX
BF6F DD2142BF          LD      IX,PIYO0
BF73 119EBE            LD      DE,SPPAT
BF76 CD82BF            CALL    PMOVS
BF79 CD82BF            CALL    PMOVS
BF7C CD82BF            CALL    PMOVS
BF7F CD82BF            CALL    PMOVS
BF82 DD3502    PMOVS:  DEC     (IX+2)
BF85 CC04C0            CALL    Z,PDCHN
BF88 CD17C0            CALL    PTCHN
BF8B DD4600            LD      B,(IX+0)
BF8E DD4E01            LD      C,(IX+1)
BF91 DD7E03            LD      A,(IX+3)
BF94 FE01              CP      1
BF96 3827              JR      C,BCSTA
BF98 280C              JR      Z,PDCK1
BF9A FE03              CP      3
BF9C 3814              JR      C,PDCK2
BF9E 2817              JR      Z,PDCK3
BFA0 05        PDCK4:  DEC     B
BFA1 CC04C0            CALL    Z,PDCHN
BFA4 1819              JR      BCSTA
BFA6 CDEFBF    PDCK1:  CALL    BICHK
BFA9 0C                INC     C
BFAA 79                LD      A,C
BFAB FE0E              CP      DMAXY-2
BFAD 3810              JR      C,BCSTA
BFAF 0D                DEC     C
```

```
BFB0 180D                   JR      BCSTA
BFB2 CDEFBF       PDCK2:    CALL    BICHK
BFB5 1808                   JR      BCSTA
BFB7 CDEFBF       PDCK3:    CALL    BICHK
BFBA 0D                     DEC     C
BFBB F2BFBF                 JP      P,BCSTA
BFBE 0C                     INC     C
BFBF DD7000       BCSTA:    LD      (IX+0),B        ・各ピーヨを適当に動かす
BFC2 DD7101                 LD      (IX+1),C
BFC5 C5                     PUSH    BC
BFC6 D9                     EXX
BFC7 C1                     POP     BC
BFC8 50                     LD      D,B
BFC9 59                     LD      E,C
BFCA 14                     INC     D
BFCB 1C                     INC     E
BFCC 3601                   LD      (HL),1
BFCE 23                     INC     HL
BFCF 70                     LD      (HL),B
BFD0 23                     INC     HL
BFD1 71                     LD      (HL),C
BFD2 23                     INC     HL
BFD3 3601                   LD      (HL),1
BFD5 23                     INC     HL
BFD6 72                     LD      (HL),D
BFD7 23                     INC     HL
BFD8 71                     LD      (HL),C
BFD9 23                     INC     HL
BFDA 3601                   LD      (HL),1
BFDC 23                     INC     HL
BFDD 70                     LD      (HL),B
BFDE 23                     INC     HL
BFDF 73                     LD      (HL),E
BFE0 23                     INC     HL
BFE1 3601                   LD      (HL),1
BFE3 23                     INC     HL
BFE4 72                     LD      (HL),D
BFE5 23                     INC     HL
BFE6 73                     LD      (HL),E
BFE7 23                     INC     HL
BFE8 D9                     EXX
BFE9 010500                 LD      BC,5
BFEC DD09                   ADD     IX,BC
BFEE C9                     RET

BFEF 04           BICHK:    INC     B
BFF0 78                     LD      A,B
BFF1 FE1F                   CP      DMAXX-1
BFF3 D8                     RET     C
BFF4 05                     DEC     B
BFF5 CD2FC0                 CALL    RND             ・画面右端へ到達した場合には最低16回以上後退す
BFF8 F610                   OR      00010000B         るようにする
BFFA E61F                   AND     00011111B
BFFC DD7702                 LD      (IX+2),A
BFFF DD360304               LD      (IX+3),4
C003 C9                     RET

C004 CD2FC0       PDCHN:    CALL    RND
C007 F602                   OR      00000010B
C009 E60F                   AND     00001111B
C00B DD7702                 LD      (IX+2),A
C00E CD2FC0                 CALL    RND             ・方向を変更し、カウンタ値を設定
C011 E603                   AND     00000011B
C013 DD7703                 LD      (IX+3),A
C016 C9                     RET

C017 08           PTCHN:    EX      AF,AF'
C018 DD7E04                 LD      A,(IX+4)
C01B EE01                   XOR     1
C01D DD7704                 LD      (IX+4),A
```

```
C020 215BBF              LD      HL,CHR00
C023 2803                JR      Z,PMPTS          ・ピーヨ番号を変更する
C025 2163BF              LD      HL,CHR01
C028 010800    PMPTS:    LD      BC,8
C02B EDB0                LDIR
C02D 08                  EX      AF,AF'
C02E C9                  RET

C02F 3E0A      RND:      LD      A,10
C031 6F                  LD      L,A
C032 87                  ADD     A,A
C033 87                  ADD     A,A
C034 85                  ADD     A,L              ・乱数発生ルーチン
C035 3C                  INC     A
C036 3230C0              LD      (RND+1),A
C039 C9                  RET

C03A □□ *0200  CKMAP:    DS      DMAXX*DMAXY      ・描き換え情報マップエリア

C23A 00010001  MAP00:    DB      0,1,0,1,0,1,0,1,0,1,0,1,0,1,0,1  ・背景①マップ
C23E 00010001
C242 00010001
C246 00010001
C24A 00010001            DB      0,1,0,1,0,1,0,1,0,1,0,1,0,1,0,1
C24E 00010001
C252 00010001
C256 00010001
C25A 00010001            DB      0,1,0,1,0,1,0,1,0,1,0,1,0,1,0,1
C25E 00010001
C262 00010001
C266 00010001
C26A 00010001            DB      0,1,0,1,0,1,0,1,0,1,0,1,0,1,0,1
C26E 00010001
C272 00010001
C276 00010001
C27A 00010001            DB      0,1,0,1,0,1,0,1,0,1,0,1,0,1,0,1
C27E 00010001
C282 00010001
C286 00010001
C28A 00010001            DB      0,1,0,1,0,1,0,1,0,1,0,1,0,1,0,1
C28E 00010001
C292 00010001
C296 00010001
C29A 00010001            DB      0,1,0,1,0,1,0,1,0,1,0,1,0,1,0,1
C29E 00010001
C2A2 00010001
C2A6 00010001
C2AA 00010001            DB      0,1,0,1,0,1,0,1,0,1,0,1,0,1,0,1
C2AE 00010001
C2B2 00010001
C2B6 00010001
C2BA 00010001            DB      0,1,0,1,0,1,0,1,0,1,0,1,0,1,0,1
C2BE 00010001
C2C2 00010001
C2C6 00010001
C2CA 00010001            DB      0,1,0,1,0,1,0,1,0,1,0,1,0,1,0,1
C2CE 00010001
C2D2 00010001
C2D6 00010001
C2DA 00010001            DB      0,1,0,1,0,1,0,1,0,1,0,1,0,1,0,1
C2DE 00010001
C2E2 00010001
C2E6 00010001
C2EA 00010001            DB      0,1,0,1,0,1,0,1,0,1,0,1,0,1,0,1
C2EE 00010001
C2F2 00010001
C2F6 00010001
C2FA 00010001            DB      0,1,0,1,0,1,0,1,0,1,0,1,0,1,0,1
C2FE 00010001
C302 00010001
```

```
C306 00010001
C30A 00010001          DB        0,1,0,1,0,1,0,1,0,1,0,1,0,1,0,1
C30E 00010001
C312 00010001
C316 00010001
C31A 00010001          DB        0,1,0,1,0,1,0,1,0,1,0,1,0,1,0,1
C31E 00010001
C322 00010001
C326 00010001
C32A 00010001          DB        0,1,0,1,0,1,0,1,0,1,0,1,0,1,0,1
C32E 00010001
C332 00010001
C336 00010001
C33A 00010001          DB        0,1,0,1,0,1,0,1,0,1,0,1,0,1,0,1
C33E 00010001
C342 00010001
C346 00010001
C34A 00010001          DB        0,1,0,1,0,1,0,1,0,1,0,1,0,1,0,1
C34E 00010001
C352 00010001
C356 00010001
C35A 00010001          DB        0,1,0,1,0,1,0,1,0,1,0,1,0,1,0,1
C35E 00010001
C362 00010001
C366 00010001
C36A 00010001          DB        0,1,0,1,0,1,0,1,0,1,0,1,0,1,0,1
C36E 00010001
C372 00010001
C376 00010001
C37A 00010001          DB        0,1,0,1,0,1,0,1,0,1,0,1,0,1,0,1
C37E 00010001
C382 00010001
C386 00010001
C38A 00010001          DB        0,1,0,1,0,1,0,1,0,1,0,1,0,1,0,1
C38E 00010001
C392 00010001
C396 00010001
C39A 00010001          DB        0,1,0,1,0,1,0,1,0,1,0,1,0,1,0,1
C39E 00010001
C3A2 00010001
C3A6 00010001
C3AA 00010001          DB        0,1,0,1,0,1,0,1,0,1,0,1,0,1,0,1
C3AE 00010001
C3B2 00010001
C3B6 00010001
C3BA 00010001          DB        0,1,0,1,0,1,0,1,0,1,0,1,0,1,0,1
C3BE 00010001
C3C2 00010001
C3C6 00010001
C3CA 00010001          DB        0,1,0,1,0,1,0,1,0,1,0,1,0,1,0,1
C3CE 00010001
C3D2 00010001
C3D6 00010001
C3DA 00010001          DB        0,1,0,1,0,1,0,1,0,1,0,1,0,1,0,1
C3DE 00010001
C3E2 00010001
C3E6 00010001
C3EA 00010001          DB        0,1,0,1,0,1,0,1,0,1,0,1,0,1,0,1
C3EE 00010001
C3F2 00010001
C3F6 00010001
C3FA 00010001          DB        0,1,0,1,0,1,0,1,0,1,0,1,0,1,0,1
C3FE 00010001
C402 00010001
C406 00010001
C40A 00010001          DB        0,1,0,1,0,1,0,1,0,1,0,1,0,1,0,1
C40E 00010001
C412 00010001
C416 00010001
C41A 00010001          DB        0,1,0,1,0,1,0,1,0,1,0,1,0,1,0,1
```

```
C41E 00010001
C422 00010001
C426 00010001
C42A 00010001               DB      0,1,0,1,0,1,0,1,0,1,0,1,0,1,0,1
C42E 00010001
C432 00010001
C436 00010001

C43A 00000000      MAP01:   DB      0,0,0,0,0,0,0,2,2,2,2,2,2,2,2,2     ・背景②マップ
C43E 00000002
C442 02020202
C446 02020202
C44A 00000000               DB      0,0,0,0,0,0,0,2,1,2,1,2,1,2,1,2
C44E 00000002
C452 01020102
C456 01020102
C45A 00000000               DB      0,0,0,0,0,0,0,2,1,2,1,2,1,2,1,2
C45E 00000002
C462 01020102
C466 01020102
C46A 00000000               DB      0,0,0,0,0,0,0,2,1,2,1,2,1,2,1,2
C46E 00000002
C472 01020102
C476 01020102
C47A 00000000               DB      0,0,0,0,0,0,0,2,1,2,1,2,1,2,1,2
C47E 00000002
C482 01020102
C486 01020102
C48A 00000000               DB      0,0,0,0,0,0,0,2,2,2,2,2,2,2,2,2
C48E 00000002
C492 02020202
C496 02020202
C49A 00000000               DB      0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
C49E 00000000
C4A2 00000000
C4A6 00000000
C4AA 00000000               DB      0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
C4AE 00000000
C4B2 00000000
C4B6 00000000
C4BA 00000000               DB      0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
C4BE 00000000
C4C2 00000000
C4C6 00000000
C4CA 00000000               DB      0,0,0,0,0,0,2,1,1,2,2,2,2,2,2,2
C4CE 00000201
C4D2 01020202
C4D6 02020202
C4DA 00000002               DB      0,0,0,2,1,1,2,0,0,2,0,0,2,2,2,2
C4DE 01010200
C4E2 00020000
C4E6 02020202
C4EA 00000000               DB      0,0,0,0,0,0,2,1,1,2,2,2,2,2,2,2
C4EE 00000201
C4F2 01020202
C4F6 02020202
C4FA 00000000               DB      0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
C4FE 00000000
C502 00000000
C506 00000000
C50A 00000000               DB      0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
C50E 00000000
C512 00000000
C516 00000000
C51A 00000000               DB      0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
C51E 00000000
C522 00000000
C526 00000000
C52A 00000000               DB      0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
C52E 00000000
```

```
C532 00000000
C536 00000000
C53A 01020102              DB      1,2,1,2,1,2,1,2,1,2,1,2,1,2,1,2
C53E 01020102
C542 01020102
C546 01020102
C54A 00000000              DB      0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
C54E 00000000
C552 00000000
C556 00000000
C55A 00000000              DB      0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
C55E 00000000
C562 00000000
C566 00000000
C56A 00000000              DB      0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
C56E 00000000
C572 00000000
C576 00000000
C57A 00000000              DB      0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
C57E 00000000
C582 00000000
C586 00000000
C58A 00000000              DB      0,0,0,0,0,0,0,0,0,0,0,0,2,2,2,2
C58E 00000000
C592 00000000
C596 02020202
C59A 00000000              DB      0,0,0,0,0,0,0,0,0,0,0,2,2,2,2,2
C59E 00000000
C5A2 00000002
C5A6 02020202
C5AA 00000000              DB      0,0,0,0,0,0,2,1,1,2,1,1,2,2,2,2
C5AE 00000201
C5B2 01020101
C5B6 02020202
C5BA 00000000              DB      0,0,0,0,0,0,2,1,1,2,1,1,2,1,1,2
C5BE 00000201
C5C2 01020101
C5C6 02010102
C5CA 00000000              DB      0,0,0,0,0,0,2,1,1,2,1,1,2,1,1,2
C5CE 00000201
C5D2 01020101
C5D6 02010102
C5DA 00000000              DB      0,0,0,0,0,0,2,1,1,2,1,1,2,1,1,2
C5DE 00000201
C5E2 01020101
C5E6 02010102
C5EA 00000000              DB      0,0,0,0,0,0,2,1,1,2,1,1,2,2,2,2
C5EE 00000201
C5F2 01020101
C5F6 02020202
C5FA 00000000              DB      0,0,0,0,0,0,0,0,0,0,0,2,2,2,2,2
C5FE 00000000
C602 00000002
C606 02020202
C60A 00000000              DB      0,0,0,0,0,0,0,0,0,0,0,0,2,2,2,2
C60E 00000000
C612 00000000
C616 02020202
C61A 00000000              DB      0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
C61E 00000000
C622 00000000
C626 00000000
C62A 00000000              DB      0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
C62E 00000000
C632 00000000
C636 00000000
C63A 00 *0010   MAP1S:     DUP     0,16             ・背景②ループ用サブエリア

C64A 00000000   MAP02:     DB      0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0   ・背景③マップ
C64E 00000000
```

```
C652 00000000
C656 00000000
C65A 00000000          DB       0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
C65E 00000000
C662 00000000
C666 00000000
C66A 00000000          DB       0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
C66E 00000000
C672 00000000
C676 00000000
C67A 00000000          DB       0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
C67E 00000000
C682 00000000
C686 00000000
C68A 00000000          DB       0,0,0,0,0,0,1,2,2,2,2,2,2,2,2,2
C68E 00000102
C692 02020202
C696 02020202
C69A 00000000          DB       0,0,0,0,0,0,1,3,3,3,3,3,3,3,3,3
C69E 00000103
C6A2 03030303
C6A6 03030303
C6AA 00000000          DB       0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
C6AE 00000000
C6B2 00000000
C6B6 00000000
C6BA 00000000          DB       0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
C6BE 00000000
C6C2 00000000
C6C6 00000000
C6CA 00000000          DB       0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
C6CE 00000000
C6D2 00000000
C6D6 00000000
C6DA 00000000          DB       0,0,0,0,0,0,0,0,0,0,0,0,1,2,2,2
C6DE 00000000
C6E2 00000000
C6E6 01020202
C6EA 02020202          DB       2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2
C6EE 02020202
C6F2 02020202
C6F6 02020202
C6FA 03030303          DB       3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3
C6FE 03030303
C702 03030303
C706 03030303
C70A 00000000          DB       0,0,0,0,0,0,0,0,0,0,0,0,1,3,3,3
C70E 00000000
C712 00000000
C716 01030303
C71A 00000000          DB       0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
C71E 00000000
C722 00000000
C726 00000000
C72A 00000000          DB       0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
C72E 00000000
C732 00000000
C736 00000000
C73A 00000000          DB       0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
C73E 00000000
C742 00000000
C746 00000000
C74A 00000001          DB       0,0,0,1,2,2,2,2,2,2,2,2,2,2,2,2
C74E 02020202
C752 02020202
C756 02020202
C75A 00000001          DB       0,0,0,1,3,3,3,3,3,3,3,3,3,3,3,3
C75E 03030303
C762 03030303
C766 03030303
```

```
C76A 00000000          DB      0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
C76E 00000000
C772 00000000
C776 00000000
C77A 00000000          DB      0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
C77E 00000000
C782 00000000
C786 00000000
C78A 00000000          DB      0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
C78E 00000000
C792 00000000
C796 00000000
C79A 00000000          DB      0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
C79E 00000000
C7A2 00000000
C7A6 00000000
C7AA 00000000          DB      0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
C7AE 00000000
C7B2 00000000
C7B6 00000000
C7BA 00000000          DB      0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
C7BE 00000000
C7C2 00000000
C7C6 00000000
C7CA 00000000          DB      0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
C7CE 00000000
C7D2 00000000
C7D6 00000000
C7DA 00000000          DB      0,0,0,0,0,0,0,0,0,0,0,0,0,1,2,2
C7DE 00000000
C7E2 00000000
C7E6 00010202
C7EA 00000000          DB      0,0,0,0,0,0,0,0,0,0,0,0,0,1,2,2
C7EE 00000000
C7F2 00000000
C7F6 00010202
C7FA 00000000          DB      0,0,0,0,0,0,0,0,0,0,0,0,0,1,3,3
C7FE 00000000
C802 00000000
C806 00010303
C80A 00000000          DB      0,0,0,0,0,0,0,0,0,0,0,0,0,1,3,3
C80E 00000000
C812 00000000
C816 00010303
C81A 00000000          DB      0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
C81E 00000000
C822 00000000
C826 00000000
C82A 00000000          DB      0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
C82E 00000000
C832 00000000
C836 00000000
C83A 00000000          DB      0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
C83E 00000000
C842 00000000
C846 00000000
C84A 00 *0010  MAP2S:  DUP     0,16              ・背景③ループ用サブエリア
```

サンプルですから、マップもパターンも最低限しか用意していませんが、なんとなくゴーストタウンを5匹の『おばけのピーヨ』が飛び回っているという雰囲気が感じられたと思います。スペースキーを押すと、画面は一時停止します。ソフト・スプライトによって背景とキャラクタ、あるいはキャラクタとキャラクタとの重ね合わせも完璧に行われていることがわかります。スクロールは、動かない背景①、1回おきに動く背景②、毎回動く背景③の3重スクロールです。

それぞれのマップは1面分しか用意してないので、データをグルグル回してス

クロールさせています。横スクロールの場合、部分描き換えを縦に行ったほうがブレが少なくなると「4-1　スクロールとキャラクタ」で書きましたが、そのためにはデータ自体を縦に並べたほうが好都合です。

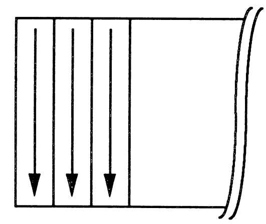


図 4-4-1　マップデータの並びと描き換えの順序

ここで、マップデータの内容についても触れておきましょう。背景①（MAP00）は一番下に位置していますからすべてベタ（透明データが不要）ですが、背景②と③（MAP01とMAP02）のパターンは3種類のタイプに分けることができます。

●全 透 明：空間を意味し下のパターンが出る＝データ不要
●部分透明：部分的に下のパターンが出る＝透明データ＋BRGデータ
●ベ　　タ：下のパターンは見えない＝BRGデータ

これらはパターン番号で区別されており、全透明はパターン番号＝0、部分透明とベタはラベル（背景①＝FPAT 1、背景②＝FPAT 2）で区別しています。ここではパターン数が少ないため、いずれも2以降がベタとなっています。ベタ・パターンは、それより下位にあるパターンを表示しなくていいので、表示速度が速い上にデータ量が少ない（透明データ不要）というメリットがあります。さらに、ベタ・パターンをG.VRAMの余りに用意しておけば（本プログラムではFE90H以降を利用)、3面同時にデータを転送することができます。

ただし、画面に直接転送して作画するのは、ベタ・パターンの上に重なるパターンがない場合で、上に別のマップが重なる場合には、いったんFE80H以降にデータを作成してから転送します。このほうが速度効率がいいことは「2-5　グラフィックVRAMの余り」で説明したとおりです。また、キャラクタを表示しなければならない場合には、ソフト・スプライト処理として、これとは別にG.VRAMの一部（本プログラムではF980H以降）に合成したパターンを前もって作成しておきます。こうすると、描き換えに要する時間が短縮され、スクロールがよりきれいになるからです。もちろん、このエリアは通常のG.VRAMですから、テキスト画面によってマスクをしておく必要があります。

したがって、部分描き換えと一口にいっても、単に前回と違うパターンを描くだけではなく、その内容によって処理方法を違えたり準備も必要ということです。この例ではキャラクタを背景②と③の間に置いていますから、実際には次のような描き換えのタイプが存在しています。

	背景①	背景②	キャラクタ	背景③
直接表示（3面転送）	☆ベタ － －	無 ☆ベタ －	無 無 無	無 無 ☆ベタ
FE80Hに作成後転送	☆ベタ ☆ベタ ☆ベタ －	☆合成 無 ☆合成 ☆ベタ	無 無 無 無	無 ☆合成 ☆合成 ☆合成
ソフトスプライト （データを事前に作成）	☆ベタ ☆ベタ ☆ベタ ☆ベタ － － －	無 無 ☆合成 ☆合成 ☆ベタ ☆ベタ －	☆合成 ☆合成 ☆合成 ☆合成 ☆合成 ☆合成 －	無 ☆合成 無 ☆合成 無 ☆合成 ☆ベタ

☆　：作画の必要なパターン
－　：作画の不要なパターン
無　：パターンが存在しない／空間
合成：重ね合わせの必要なパターン

これらは描き換えの際にすぐ判別できるよう、予め調べておくことが大切です。さもないと、描き換え時間が長くなりスクロールが汚くなってしまいます。そのため、こういった描き換え情報を示す専用のマップ（CKMAP）を用意し、ソフト・スプライトのデータ同様事前に作成しておきます。その内容は図 4-4-2のようなものです。

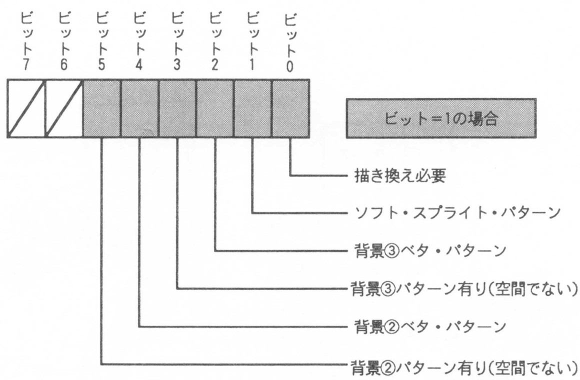


図 4-4-2　描き換え情報データ

データはスクロールの有無にかかわらず毎回更新します。というのは、ソフト・スプライトの情報もこの中に含まれるからです。更新は、背景②→背景③の順で行い、描き換え不要の場合でもその他のチェック（ベタかどうか／パターンの有無）をしておかなければなりません。次に、前回ソフト・スプライトのあった座標はすべて描き換え必要の判定をします。ここでキャラクタ移動（座標変化など）の処理をし、新たな座標について描き換え必要及びソフト・スプライトの認定をします。そして、ソフト・スプライト用のデータを作成すれば、すべての情報は

整ったことになり、それを基に必要な表示を行えばいいわけです。

しかし、このソフト・スプライト用データ作成という作業は、簡単そうで意外と面倒なものです。今回は20のソフト・スプライトを利用できるようにしていますが、キャラクタの座標が変化するたびに当然この座標も変化します。一方、描き換えは図4-4-1の順序で行いますから、データは表示に合わせた並びで作成しておく必要があります。プログラム中にあるMLANKが座標に順位を付けるルーチンで、データ作成はそこで付けられた順位を参照しながら行うようになっています。

ここではすべてのソフト・スプライトを利用していますが、常にすべてが利用されるとは限りません。そこで、利用しないものの順位は0としてデータ作成などから除外するようにしています。また、同じ座標に複数のソフト・スプライトがあった場合には、番号の若いほうが上位に表示されるようプログラムで優先順位を付けています。

キャラクタの動きなどについては簡単ですから特に解説しませんが、移動方向は0＝停止、1＝右下、2＝右、3＝右上、4＝左となっており、乱数によって適当に動かしています。実際のゲームでは巨大キャラなども登場しますが、こういう場合にはソフト・スプライト側でもベタ・パターンと重ね合わせパターンとに分けるようにするなど、まだまだプログラムには発展の余地があります。この際ですから、とことんそういった可能性を追求してみてください。



【マップの裏話】

画面を構成するマップはデータを大量に消費します。一概にマップが広ければいいとは思いませんが、マップが広大になればゲームも大きくできるのは事実です。本章においてマップデータを工夫するアイデアを解説したのも、このような理由からです。
ところで、よくゲームの広告に画面数（マップ数ではない）が何万と書いてあるものがあります。まともにこんなことをしたら、それだけでディスクが十枚以上必要になってしまいます。実はこれ、同じ構成のマップをアチコチで使うということなのです。例えばの話、4画面で構成されているマップを10回利用すれば40画面……となるわけです。
個人的にはこういうマップは嫌いですし、データの工夫とは認められません。第一、同じ構成のマップを違う場所と判断しなければならないなんて不自然です。トラップとして利用したり、距離感を出すために数回ループする程度でないと、プレイ中にイライラしてしまいそう……。こういった計算が成立するなら、先ほどのサンプル・スクロールなんて画面数は無限ですからネ。

【スポット処理】

テキスト画面を用いたスポット処理を第3章で紹介しましたが、グラフィック画面だけで完全なスポット処理を実現しているゲームも見かけます。難しそうですが、多重スクロールの背景②の役目をもう一度思いだしてください。背景②は下にあるパターンの上に合成するわけですから、これを黒いパターンだけで作れば簡単にスポットを付けることができますね。それどころか、星型でもハート型でも構わないし、スポットの周囲をボカすような本格的スポットだって可能なのです。
な～ンだ……なんてバカにしてはいけません。こういったことは、手品と同じでタネを知る前に考えてこそ価値があるのですから……。

【1バイト以下のスクロール】

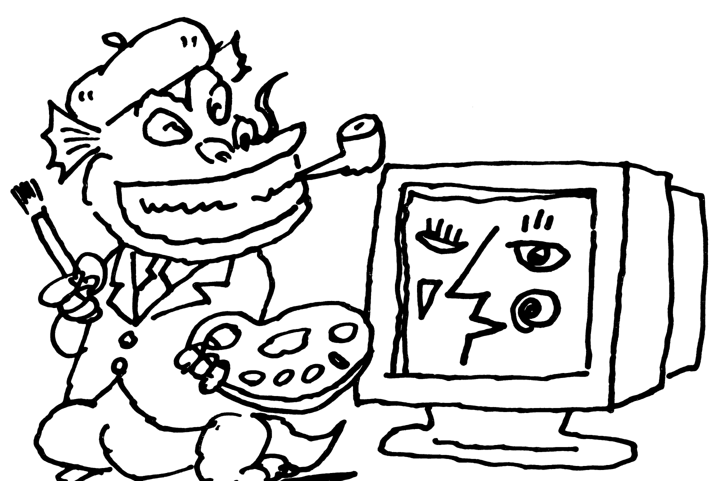
最後の節の多重スクロールでは1回おきに動く背景②というのがありましたが、これを毎回1バイト単位で動かすとスクロールはより美しくなります。このような場合、マップを小さなパターン（8×8ドット）にしたのではマップデータも2倍になり処理も大変です。そこで、プログラムでマップを半分ずつ（右半分と右隣りの左半分）表示できるようにしておくと、毎回1バイト単位でスクロールさせることができます。
プログラムは常に進化していないと飽きられてしまいます。多重スクロールの基本を理解したなら、そこに新たなアイデアを盛り込むのは当然のこと。それがプログラムの個性です。本書のプログラムがどのように飛躍するのか……、マシン語の世界には夢とロマンがあふれているのです。

第5章　魅惑の大型グラフィックス

いわゆるCG（コンピュータ・グラフィックス）とアクション・ゲームで使われるパターン画との境界線はどこにあるのでしょうか。どちらも同じグラフィック画面に描かれた絵であり、違いを明確に規定するものはありません。

とはいえ、CGはデータをメモリに保存し切れないほど大きい絵……というばく然とした認識はあるようで、一昔前にはCGといえばBASICのLINE, PAINT,PSET命令によって描かれるのが普通でした。これが徐々にマシン語化され、高速ラインルーチンとか高速ペイントルーチンなどが開発されたのです。このままの状態が続けば、パターン画はG.VRAMに直接データを入れるもの、CGはライン／ペイントで描くものという定義ができていたかもしれません。

……が、それがすでに過去の話であることは明らかです。いまでは、ライン／ペイントでは作画不可能なドット絵も登場し、しかも当然のように高速に表示されています。そこにどのような進歩があったのか、過去をちょっぴり気にしながら身近になったCGをエンジョイしちゃいましょう。



5-1 スキャナによる画像入力

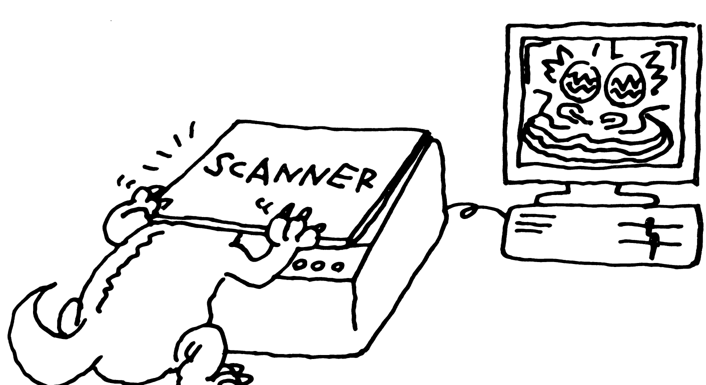
実際の表示方法がライン／ペイントであれ何であれ、大きな絵を画面に描く前にはそれなりに原画（下絵）を準備するのが普通です。いきなり真っ黒な画面に向かうのは、どんなに優秀なグラフィック・ツールを使おうと、全体の構図がつかみにくく作業効率は低下します。したがって、原画を描くということは昔も今も変わらぬCGの基礎工程といえるでしょう。

次に、それに基づいて画面上にアウトラインを描写するわけですが、ハードウェア的なサポートがないころは、透明ラップに原画を写してディスプレイに貼り付け、それをなぞるようにデジタイズしたものです。非常に原始的な方法ですが、これでも方眼紙に描いた絵から座標を求めるよりは楽だったのです。

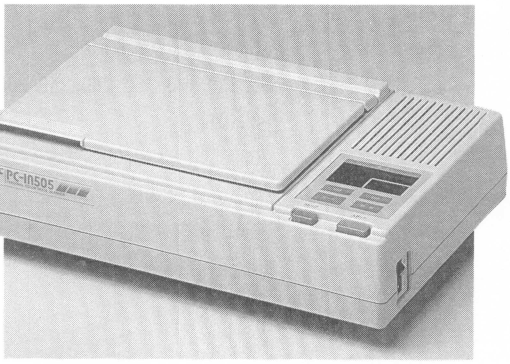
その後、色々な画像入力装置（デジタイザーなど）が開発されましたが、いま最も注目を浴びているのがイメージスキャナです。これは、写真や絵をコピー機のようなもので読み取り、それを紙ではなく画面上に複写するものです。発売当初はかなり高価な商品だったため、個人で所有するには相当勇気が要りましたが、最近は白黒専用の普及品から本格的なカラースキャナまで種類も豊富に各社から発売されるようになりました。

こういった状況から、ゲームで使用されるCGには【原画】→【スキャナ取り込み】→【ドット修正／着色】という作画工程が確立され、デジタイズの苦労は大幅に軽減されたのです。本書でも、この工程を前提にCGの解説をしていきますが、本章最後に紹介するグラフィック・ツール単独でも従来方式の作画は可能です。ただし、スキャナが1枚の絵を数十秒で取り込むのに対し、数時間の労力を要することを覚悟してください。

では、現在（1989年秋）どのようなスキャナが市販されているのか、いくつかの製品を見ながら特徴を比較してみましょう。いずれの機種も特別なインターフェースボードは必要とせず、RS-232Cのインターフェースに接続するだけで使用できます。

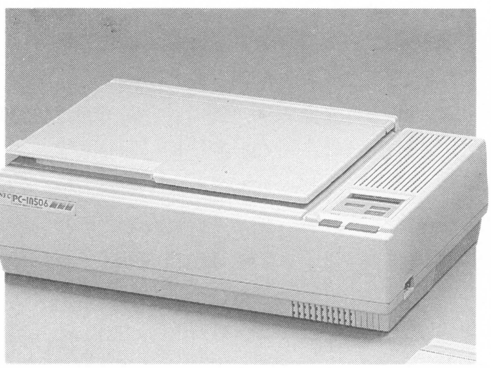


◆ PC-IN505 ◆ NEC（日本電気） ¥138,000（税別）



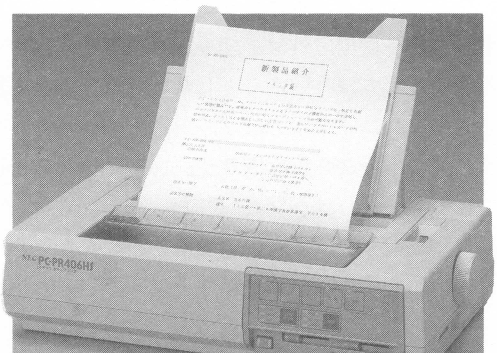
読み取り方式	：原稿固定型平面操作方式
読み取りサイズ	：A6判（最大105×160mm）
読み取り線密度	：90～400 DPI（10DPIステップ） 210DPI以上は200DPIのデータを拡大処理
階調	：2値、疑似中間調(ディザ 1,2,3) 中間調（2,4,8,16,32,64階調）
カラー読み取り	：RGB 面順次
外形	：340(W)×220(D)×100(H)mm

◆ PC-IN506 ◆ NEC（日本電気） ¥228,000（税別）



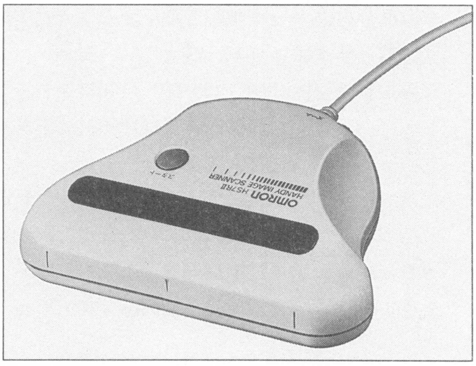
読み取り方式	：原稿固定型平面操作方式
読み取りサイズ	：A4判（最大203×297mm）
読み取り線密度	：5～320 DPI（1 DPIステップ）
階調	：2値、疑似中間調（ディザ 1,2） 中間調（2,4,8,16,32,64階調）
カラー読み取り	：RGB 面順次／RGB 線順次
外形	：500(W)×360(D)×146(H)mm

◆ PC-PR406HS ◆ NEC（日本電気） ¥138,000（税別）



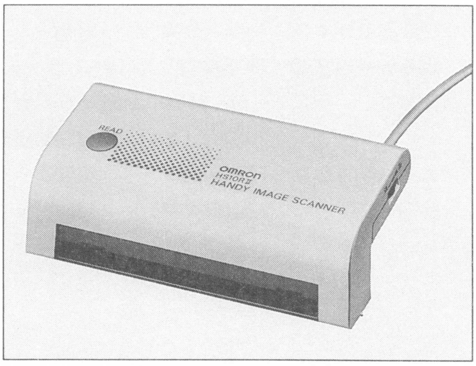
読み取り方式	：原稿移動型シリアル走査方式
読み取りサイズ	：B4判(最大205×297mm)
読み取り線密度	：90,160,180 DPI
階調	：2値、疑似中間調（ディザ） 中間調(4,8,16階調)
プリンタ部	：日本語カラー熱転写
外形	：410(W)x246(D)x103(H) mm

◆ HS7RII ◆　　オムロン（立石電機）¥39,800（税別）



読み取り方式	：スキャナ手動走査方式
読み取りサイズ	：A6幅（最大105mm）
読み取り線密度	：90, 120, 160, 180, 200, 240, 300, 400 DPI 200DPI以外は200DPIのデータを縮小／拡大処理
階調	：2値、疑似中間調（ディザ）
外形	：146(W)x111(D)x40(H)mm

◆ HS10RII ◆　オムロン（立石電機）¥49,800（税別）



読み取り方式	：スキャナ手動走査方式
読み取りサイズ	：A6幅（最大104mm）
読み取り線密度	：90, 120, 160, 180, 200, 240, 300, 400 DPI 200DPI以外は200DPIのデータを縮小／拡大処理
階調	：2値、疑似中間調（ディザ／誤差拡散）
外形	：130(W)x70(D)x41(H)mm

コピー機型、プリンター一体型、ハンディ型と、本体タイプの違いがそれぞれ読み取り方式の違いになっています。コピー機型以外は、読み取り位置を一定させることが難しいので、同じ絵を階調を変えて読み取ろうとした場合など、ある程度のズレが生じるのは避けられません。ユニークなのはプリンタと一体になっているPC-PR406HSで、ハガキサイズのものなら読み取りデータを記憶し、そのままプリントアウトすることができます。これはパソコンをまったく介さないので、スキャナ機能を利用した簡易コピー機といってもいいでしょう。ただし、読み取りはプリンタ用紙を送るように行いますから、本や箱などに印刷されたものを読み取ることはできません。

読み取り線密度のDPI（Dot Per Inch）というのは、1インチ（約25mm）を何ドットに分解して読み取るかという単位です。例えば、200DPIは1インチを200ドットに分解することですから、1ミリ当りに換算すると約8ドットです。グラフィック画面は横640ドットですから、読み取った通りに表示すれば画面幅いっぱい（640ドット）が約8cmに相当するわけです。現在普及しているディスプレイは14

インチ（対角線長）のものが主流ですから、感覚的には8cmが24～25cmに拡大表示されたことになります。したがって、原画のサイズイメージを生かすならば、70DPI前後が最適ということになりますが、原画を描く前にサイズ合わせをして感覚を把握しておくといいでしょう。

階調の2値とは、原画を完全に白／黒で識別するもので、中間色（灰色など）は濃度によって白か黒のどちらかになります。これをカバーするのが疑似中間調で、ディザ法によって濃淡を表現します。ディザ法については、各製品のマニュアルに詳しい説明がなされていますから、ここでは省きます。要するに、ドット密度によって濃淡を表現する方法のことで、新聞の白黒写真を虫メガネで拡大したようなものです。

一方、中間調識別機能があるものは、濃淡をレベルで識別することができます。例えば8階調の場合であれば、白～灰色（6レベル）～黒と8レベルに分けて読み取ることができるわけです。ただし、PC-8801の場合は8色しか表現できませんから、実際にはこの機能をカラーで使いこなすことはできません。

なお、カラー読み取りのできるPC-IN505／PC-IN506にはRGB面順次とRGB線順次という区別があります。これはカラーをプレーン順（G→R→B）に3回読んでいくか、1ラインずつ3回読んでいくかという違いです。構造的にキャリッジ（読み取り部）が3往復する面順次方式では、微妙にドットカラーがズレる危険性がありますが、線順次なら確実にカラーを捕らえることができます。ただし、640×200ドット程度ではその心配はないと考えていいでしょう。現在は価格的なことから面順次方式が普及していますが、将来は線順次方式が主流になるといわれています。

さて、スキャナの種類やだいたいの特徴は理解できたと思いますが、実際にスキャナを使うためにはスキャナからのデータを受け取るプログラムが必要です。つまり、RS-232Cインターフェースを通じて送られてきたデータを受け取り、それをG.VRAMに入れるというプログラムです。

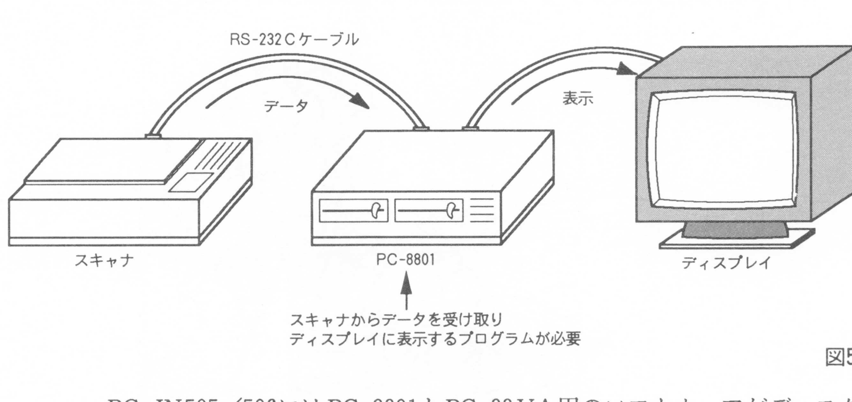


図5-1-1

PC-IN505／506にはPC-9801とPC-88VA用のソフトウェアがディスクで提供されていますが、それ以外の製品にはソフトウェアのサポートがありません。一応プログラムを組むための情報が各マニュアルに載っていますが、マシン語に相

当精通していないと自分でプログラムを組むのは難しいかもしれません。というのは、一口にデータの受け取りといっても、実体はスキャナという外部機器をコントロールすることだからです。スキャナの仕様を理解するだけでなく、RS-232Cを通してコマンドやデータを送受信できるだけの知識も要求されています。それに、ゲームに直接関与しないことを、ここで長々と解説するだけのゆとりもありません。

それならば、そういった細かなことを解説するより、使いやすいツールにして提供したほうがいい……ということで、本書ではすべての製品に使用できるPC-8801専用のソフトウェア『SCAN88』を用意しました。詳しい操作方法は最後の節で解説しますが、単にデータをG.VRAMに入れるだけでなく、次のような特殊機能が追加されています。

①表示方向の縦／横指定

スキャナで読み取ったデータを、縦方向（上から下）でも横方向（左から右）でも表示させることができます。この機能により、読み取り幅の小さいスキャナでも、横長の原画を読み取れることになります。

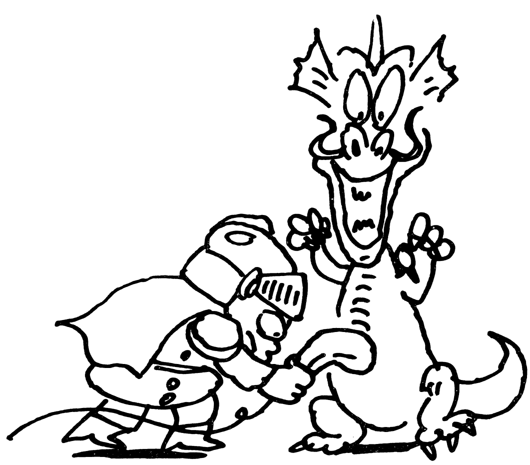
②1/2縮小表示

14インチディスプレイの場合、90DPIが最小では原画を拡大したイメージでしか表示できません。1/2縮小表示機能を利用することにより、表示サイズを細かく設定できるようになります。

③カラースキャナへの対応

将来は価格も安価なハンディ・カラースキャナが登場することが期待されます。そういったケースにもツールとして困ることがないよう、モノクロ、カラー（面順次、線順次）すべてに対応できるようにしてあります。

こうして取り込まれた画像は、即座にグラフィック・ツール『PMAN88』でエディットできるよう環境を整えてあります。もちろん、作成した絵は保存／再現が自由自在です。その秘密は……。次の節に書いてあります。



5－2　グラフィック・データの圧縮と展開

オモチャ箱をひっくり返したような乱雑な部屋でも、その気になって整理をすれば案外すっきりとまとまるものです。では、ここでいう「整理」とは、いったい何をどうすることなのか……なんて難しく考えながら片付ける人はいないでしょう。たいていは無意識のうちに同じ種類のものをまとめているはずです。

しかし、大人と子供では整理能力が違うように、バラバラの中から共通項を発見するというのは意外と知的作業なのです。例えば、本箱に本をしまうにしても、幼児はただ並べるだけですが、やがて種類別に分けることを覚えます。もう少し大きくなると、発行月順に並べたり、背の順に並べたり、あるいはよく読む順に並べたり……と、かなり個人差が現れてきます。こうなると、一概にどれが最善の整理法ということはできません。

つまり、共通項というものは必ずしも一定したものではなく、アイデアと状況により変化するものなのです。グラフィック画面をフルに使えば、80(横)×200(縦)×3(プレーン)＝48000バイトという大量のデータがG.VRAMに納まっていることになります。このままデータ化すれば、0～BB7FH番地までがデータ領域になってしまいます。そこで、共通項を捜してデータをまとめ、表示する時に元の姿に展開しようというのです。これがデータ圧縮／展開の基本です。

問題は、本の並べ方の例でもわかるように、すべてに最適となる圧縮法はないということです。ある絵に対しては強力な圧縮がかかるのに、別の絵に対しては圧縮どころか逆にデータが増えてしまうということだってあるのです。次のデータ圧縮法は、連続した同じ数値に対し圧縮をかけるという、最も基本的な圧縮サンプルです。

圧縮前のデータ：55　55　55　55　55　12　FF　FF　FF　FF
圧縮後のデータ：55　55　03　12　FF　FF　02

ここでの圧縮サインは同じ数値の連続です。同じ数値が2つ続いたら、次の数値でその値があといくつ連続するかを示しています。この圧縮法により、この例では10個のデータが7個になっています……が。

圧縮前のデータ：55　55　FF　FF　55　55　FF　FF　55　55
圧縮後のデータ：55　55　00　FF　FF　00　55　55　00　FF　FF
　　　　　　　　00　55　55　00

今度は、圧縮どころかデータが増えてしまいました。圧縮には、こういった危険性が常に潜んでいるのです。ご存じの方もいるかもしれませんが、以前ある雑誌で『グラフィック・データ圧縮コンテスト』というものを企画したことがあります。それは複数の課題絵を提供して圧縮率を競い合うというものですが、コンテストですから圧縮法は応募者によってすべて違います。その結果、絵によって

順位はバラバラになることが事実として確認されたのです。これは、ある程度予測していたことでしたが、これほどハッキリと絵の影響を受けるとは思っていませんでした。

しかし、「だから汎用の圧縮法は存在しない」というのではありません。この事実が示したことは、「総合して優秀なものが最も汎用性の高い圧縮ルーチンである」ということの裏返しなのです。これから紹介する圧縮法は、私が全力を投入して開発したバージョンを一部（かなり）カットしたものですが、それでも十分に実用に耐えるものです。カットした理由は、ページ数の問題が最大のものですが、さらに独自のアイデアを追加してほしいという希望もあったからです。それがなければ圧縮ルーチンをソースリストで紹介する意味がなくなってしまうでしょう。

なお、ここでの圧縮は数ある圧縮法の中の一例であり、すべての圧縮が似たような考え方に基づいているというわけではありません。例えば、スキャナを購入するとマニュアルには多段階圧縮法というビット単位の圧縮の説明がありますが、データをまったく違う視点からまとめています。ただ、多段階圧縮はカラーグラフィックスに適した圧縮ではないため、そのままゲーム等に応用するのは有効とはいえません。とはいえ、この考え方を基本に独自の工夫改良を加えてゲーム用の圧縮とした例もありますから、興味のある方はスキャナのマニュアルをよく読んで参考にするといいでしょう。

では、本書の圧縮法で使われている圧縮データの構造を示します。データはG.VRAMの内容を1バイトづつ縦方向に読んでいますが、これは画面が絵であることから横方向より縦方向に共通性が高いからです。よくわからない場合は、図5-2-1からどちらのペン先が絵を描きやすいか考えてください。

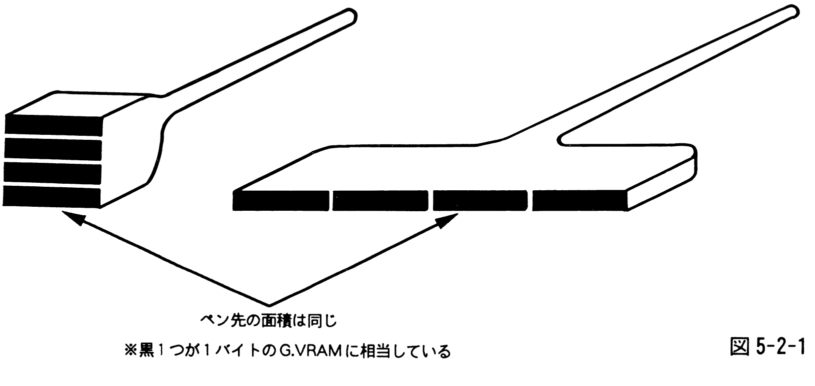


図5-2-1

一般的な圧縮の基本は前後のデータとの共通性ですが、グラフィック画面の場合は前後のデータだけでなく、他のプレーンのデータも比較の対象となります。実は、これがグラフィック・データ圧縮の最大のポイントで、単に他のプレーンと同じかどうかを調べるだけでもかなりの効果があります。さらに、他のプレーンデータをANDやXORを取ってから調べると、そこには複数の共通性が存在していることもあり、どれを選択するかはまさに思考ゲームといっても過言ではあり

ません。ここでは、そういったすべての可能性を追求してはいませんし、複数の共通性の中から必ずしも最良の選択がなされているとも限りません。この先、どこまでアイデアと効率を追求できるか、グラフィック・データの圧縮はここからが真のスタートといえるでしょう。もちろん、私自身も新たなアイデアとアルゴリズムによる構想は完了しています。あとは、それをプログラミングするきっかけが必要なだけです。そのきっかけとは……、読者の皆さんによって私の圧縮法が過去形になることです。オリジナルな圧縮法の発見／開発の朗報をお待ちしています。

それでは、本プログラムでの圧縮データの構造と、圧縮のサインを紹介しましょう。

圧縮データの構造

圧縮データは2000H～7FFFH番地に生成され、それを越えるデータは無視されます。展開する際は、C000H番地以前であればどこにデータを置いても構いません。

2000H　：プレーン別にデータの有無、A／B圧縮の区別をする

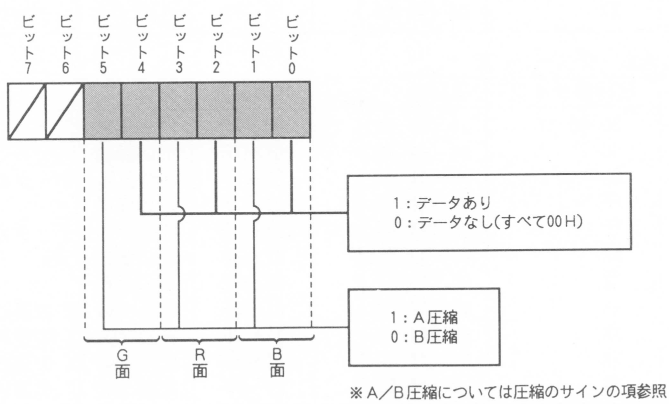


図5-2-2

2001H　：横バイト数

2002H　：縦ライン数

2003H～：圧縮されたデータ

圧縮のサイン

圧縮のサインはデータの上位／下位が同じ数値の場合（A圧縮＝すべて、B圧縮＝00, FF, 55, AA, 33, 66, 99, CCのみ）で、その次に連続するデータ数が示されます。ポイントはこの連続数の数え方で、一定の範囲内で数えることによって圧縮の内容に変化をつけています。

例えば、連続数の範囲（□□の部分）が81～FFとなっていれば、ビット7が圧縮の内容を示すフラグで、1～7Fがその連続する数というわけです。もし、連続する数が□□の範囲を越える場合は、□□をその範囲での最大値とした上で、次の2バイト（## ##の部分）で連続する総数を表します。

XとYは1バイトの数値の上位あるいは下位4ビットのことで、XX＝上位／下

位が同じ数値、XY＝上位／下位が違う数値という意味です。また、データ（XX）がローテートする場合は（※）で示し、55, AAについては1ローテート、それ以外の数値は2ローテートします。

★B（ブルー）面

圧縮のサイン	連続数	□□の範囲	内容
00 00 XY	□□（## ## ）	04～FF	XYの連続
00	□□（## ## ）	01～FF	00の連続
FF	□□（## ## ）	01～FF	FFの連続
XX	□□（## ## ）	01～7F	XXの連続（※）
XX	□□（## ## ）	81～FF	XXの連続

★R（レッド）面

圧縮のサイン	連続数	□□の範囲	内容
00 00 XY	□□（## ## ）	04～FF	XYの連続
00	□□（## ## ）	01～7F	00の連続
00	□□（## ## ）	81～FF	B面と同じデータの連続
FF	□□（## ## ）	01～FF	FFの連続
XX	□□（## ## ）	01～7F	XXの連続（※）
XX	□□（## ## ）	81～AF	XXの連続
XX	□□（## ## ）	B1～BF	B面 AND XX の連続
XX	□□（## ## ）	C1～FF	B面 AND XX の連続（※）

★G（グリーン）面

圧縮のサイン	連続数	□□の範囲	内容
00 00 XY	□□（## ## ）	04～FF	XYの連続
00	□□（## ## ）	01～7F	00の連続
00	□□（## ## ）	81～FF	B面と同じデータの連続
FF	□□（## ## ）	01～7F	FFの連続
FF	□□（## ## ）	81～FF	R面と同じデータの連続
XX	□□（## ## ）	01～3F	XXの連続（※）
XX	□□（## ## ）	41～5F	XXの連続
XX	□□（## ## ）	61～6F	B面 AND XX の連続
XX	□□（## ## ）	71～7F	R面 AND XX の連続
XX	□□（## ## ）	81～BF	B面 AND XX の連続（※）
XX	□□（## ## ）	C1～FF	R面 AND XX の連続（※）

表を見ただけでは圧縮の特徴がわからないかもしれませんが、これは主にグラフィックスのペイントされている部分に共通性を求めたものです。つまり、ゲーム等に使用されている一般的なグラフィックスに対して効果的な圧縮ということです。また、R／G面については他面のデータを参照してデータを作成することができますから、実際にはペイント以外の部分に対しても圧縮をかけていることになります。参考までに圧縮のサンプルを載せておきますので、表と照らし合わせながらデータ構造を確認してください。

圧縮の例

	圧縮前		圧縮後
B面	FF, FF, FF, FF, FF, F0, F0, F0, F0, F0, F0, 73	→	FF, 05, 00, 00, F0, 06, 73
R面	55, AA, 55, AA, 55, A0, 50, A0, 50, A0, 50, 22	→	55, CC
G面	33, 33, 33, 33, 33, F0, F0, F0, F0, F0, F0, 73	→	33, 65, 00, 87

この場合、トータルして36のデータが13に圧縮されていますから、約36％の圧縮率ということになります。ここまで圧縮の特徴を知ると、きっと色々な疑問や

アイデアが湧いてくるかもしれません。ただ、圧縮も他のプログラム同様、推定結果と実行結果が違うことがよくあります。いいと思って投入したアイデアが、結果的に役に立たないとわかるのはプログラムを組んでからの話です。疑問やアイデアは机上の空論に終わることなく、ぜひプログラムとして自分の目で結果を確認してください。おそらく、それは単に圧縮理論の向上になるだけでなく、プログラミング技術が飛躍するチャンスとなるでしょう。

なお、ここで作成するプログラム（LIST 5.1）は、裏RAMの100H番地からの圧縮／展開プログラムと、それをBASICからコールできるようにしたB700H番地からのプログラムとに分かれています。このうち、圧縮／展開プログラムは次の節で紹介するグラフィック・ツールにおいて実際に画面のロード／セーブ用としても使用しますから、必ずファイル名「PRES88」でセーブされていなければなりません（添付ディスクにはセーブされています）。

また、オリジナルな圧縮／展開プログラムを作成した場合でも、先頭部分（ジャンプテーブルとワークエリア）さえ同じであれば、同様のファイルでグラフィック・ツールに組み込むことができます。

LIST 5.1

```
                          ORG     100H

0100 C31501               JP      DPRES            ・圧縮プログラムへ
0103 C38A0B               JP      OPEN3            ・展開／表示プログラムへ

0106 00C0       PTOP:     DW      0C000H           ・圧縮画面の左上アドレス
0108 50         XSIZE:    DB      80               ・圧縮の横バイト数
0109 C8         YSIZE:    DB      200              ・圧縮の縦ライン数
010A 0000       DTADR:    DW      0                ・圧縮データエンドアドレス

010C 0020       DDTOP:    DW      2000H            ・展開／表示データ先頭アドレス
010E 00C0       XYPOS:    DW      0C000H           ・展開／表示の左上アドレス

2000=           DTOP:     EQU     2000H            ・圧縮データ先頭アドレス
0110 0000       ATOP:     DW      0                ・A圧縮のデータエンド
0112 0000       BTOP:     DW      0                ・B圧縮のデータエンド
0114 00         DCONT:    DB      0                ・データ圧縮中のサイン

0115 3E3F       DPRES:    LD      A,00111111B     |
0117 320020               LD      (DTOP),A        |
011A 3A0801               LD      A,(XSIZE)       |・データエリア初期化
011D 320120               LD      (DTOP+1),A      |
0120 3A0901               LD      A,(YSIZE)       |
0123 320220               LD      (DTOP+2),A      |
0126 210320               LD      HL,DTOP+3       }・データエンドアドレス初期化
0129 220A01               LD      (DTADR),HL      
012C 213602               LD      HL,BPRES        |
012F 118D02               LD      DE,TOBPR        |・B面を圧縮
0132 3E5C                 LD      A,5CH           |
0134 CD4E01               CALL    PDATA           |
0137 21CD02               LD      HL,RPRES        |
013A 11A503               LD      DE,TORPR        |・R面を圧縮
013D 3E5D                 LD      A,5DH           |
013F CD4E01               CALL    PDATA           |
0142 214505               LD      HL,GPRES        |
```

```
0145 110C07                LD     DE,TOGPR
0148 3E5E                  LD     A,5EH              ・G面を圧縮
014A CD4E01                CALL   PDATA
014D C9                    RET

014E 222F02      PDATA:    LD     (POINT+1),HL
0151 ED536E02              LD     (RETAD+1),DE
0155 32FF01                LD     (PLANE+1),A
0158 CDA701                CALL   TATEA              ・A圧縮を実行し無データ（すべて00H）なら
015B 3015                  JR     NC,NDATA             NDATAへ
015D CDC301                CALL   TATEB
0160 2A1001                LD     HL,(ATOP)
0163 ED5B1201              LD     DE,(BTOP)
0167 B7                    OR     A
0168 ED52                  SBC    HL,DE              ・B圧縮を実行し、A圧縮と比較していいほうを
016A DCA701                CALL   C,TATEA              採用する
016D ED530A01              LD     (DTADR),DE
0171 C9                    RET
0172 210020      NDATA:    LD     HL,DTOP
0175 3AFF01                LD     A,(PLANE+1)
0178 FE5E                  CP     5EH
017A 280A                  JR     Z,RBIT4
017C FE5D                  CP     5DH
017E 2803                  JR     Z,RBIT2
0180 CB86                  RES    0,(HL)             ・無データの場合に先頭データのフラグで示す
0182 C9                    RET
0183 CB96        RBIT2:    RES    2,(HL)
0185 C9                    RET
0186 CBA6        RBIT4:    RES    4,(HL)
0188 C9                    RET

0189 B7          ABCHK:    OR     A
018A C8                    RET    Z
018B FEFF        ABCK2:    CP     0FFH
018D C8                    RET    Z
018E FE55        CHKUD:    CP     55H
0190 C8                    RET    Z
0191 FEAA                  CP     0AAH
0193 C8                    RET    Z
0194 FE33        CKUD2:    CP     33H
0196 C8                    RET    Z
0197 FE66                  CP     66H
0199 C8                    RET    Z              ・A／B圧縮について上位／下位が同じか否かを調べ
019A FE99                  CP     99H              る
019C C8                    RET    Z
019D FECC                  CP     0CCH
019F C8                    RET    Z
01A0 0F          UDPON:    RRCA
01A1 0F                    RRCA
01A2 0F                    RRCA
01A3 0F                    RRCA
01A4 BB                    CP     E
01A5 7B                    LD     A,E
01A6 C9                    RET

01A7 3EEE        TATEA:    LD     A,0EEH
01A9 321402                LD     (SBIT5+1),A
01AC 3ECE                  LD     A,0CEH
01AE 320C02                LD     (SBIT1+1),A
01B1 3EDE                  LD     A,0DEH
01B3 321002                LD     (SBIT3+1),A
01B6 211001                LD     HL,ATOP            ・A圧縮のための初期
01B9 112C03                LD     DE,TYPRA             SET命令に書き換え
01BC 012206                LD     BC,TYPGA             0FH = RRCA
01BF 3E0F                  LD     A,0FH
01C1 181A                  JR     TATE

01C3 3EAE        TATEB:    LD     A,0AEH
01C5 321402                LD     (SBIT5+1),A
01C8 3E8E                  LD     A,8EH
```

```
01CA 320C02            LD     (SBIT1+1),A
01CD 3E9E              LD     A,9EH
01CF 321002            LD     (SBIT3+1),A       ・B圧縮のための初期化
01D2 211201            LD     HL,BTOP            RES命令に書き換え
01D5 117C03            LD     DE,TYPRB           0C9H = RRCA
01D8 01CA06            LD     BC,TYPGB
01DB 3EC9              LD     A,0C9H

01DD 32A001   TATE:    LD     (UDPON),A
01E0 22F201            LD     (TATE4+1),HL
01E3 22840B            LD     (EXIST+2),HL
01E6 ED532A03          LD     (TYPER+1),DE
01EA ED432006          LD     (TYPEG+1),BC
01EE 2A0A01            LD     HL,(DTADR)        ・プログラム／ワークエリア初期化
01F1 221001   TATE4:   LD     (ATOP),HL
01F4 AF                XOR    A
01F5 321401            LD     (DCONT),A
01F8 327E0B            LD     (DTCK0+1),A
01FB 210020            LD     HL,DTOP
01FE 3E5C     PLANE:   LD     A,5CH
0200 4F                LD     C,A
0201 ED79              OUT    (C),A
0203 FE5E              CP     5EH
0205 280C              JR     Z,SBIT5
0207 FE5D              CP     5DH               ・先頭データにA／B圧縮のフラグをセット
0209 2804              JR     Z,SBIT3
020B CBCE     SBIT1:   SET    1,(HL)
020D 1806              JR     TPRESS
020F CBDE     SBIT3:   SET    3,(HL)
0211 1802              JR     TPRESS
0213 CBEE     SBIT5:   SET    5,(HL)

0215 D9       TPRESS:  EXX
0216 2A0601            LD     HL,(PTOP)
0219 115000            LD     DE,80             ・データ圧縮開始
021C AF                XOR    A                  裏HL＝圧縮G.VRAMアドレス－80
021D ED52              SBC    HL,DE              裏B＝縦ライン数＋1
021F 4F                LD     C,A                裏C＝横バイト数カウンタ
0220 3A0901            LD     A,(YSIZE)          HL＝データアドレス
0223 3C                INC    A                  D＝前のデータを示す
0224 47                LD     B,A                E＝現在のデータを示す
0225 D9                EXX                       B＝ANDフラグ／データが同じ回数
0226 2A0A01            LD     HL,(DTADR)         C＝ANDデータ
0229 010000            LD     BC,0
022C 1600              LD     D,0
022E C33602   POINT:   JP     BPRES

0231 0600     BNORM:   LD     B,0
0233 73                LD     (HL),E            ・通常のデータ処理（B面）
0234 23                INC    HL
0235 53                LD     D,E
0236 CD300B   BPRES:   CALL   TDATA
0239 5F                LD     E,A
023A B7       BPRE4:   OR     A
023B 283E              JR     Z,BPR0
023D FEFF              CP     0FFH
023F 283A              JR     Z,BPR0            ・データの上位／下位が共通の場合それぞれの
0241 FE55              CP     55H                ルーチンへ
0243 2854              JR     Z,BPR1
0245 FEAA              CP     0AAH
0247 2850              JR     Z,BPR1
0249 CD9401            CALL   CKUD2
024C 284E              JR     Z,BPR2
024E BA                CP     D                 ・前回とデータが違えばBNORMへ
024F 20E0              JR     NZ,BNORM
0251 04                INC    B
0252 78                LD     A,B
0253 FE03              CP     3                 ・同じデータの連続が3回以下ならBNORM＋2へ
0255 20DC              JR     NZ,BNORM+2
0257 E5       SAMST:   PUSH   HL
```

```
0258 3604               LD     (HL),4
025A 2B                 DEC    HL
025B 73                 LD     (HL),E
025C 2B                 DEC    HL
025D 3600               LD     (HL),0
025F 2B                 DEC    HL
0260 3600               LD     (HL),0        ・同じデータが4回以上連続した場合の処理
0262 E1                 POP    HL
0263 010000             LD     BC,0
0266 CD7502             CALL   DCON1
0269 CD300B    SAMSL:   CALL   TDATA
026C BB                 CP     E
026D C28D02    RETAD:   JP     NZ,TOBPR
0270 CDBE0A             CALL   INC1F
0273 18F4               JR     SAMSL

0275 3E01      DCON1:   LD     A,1
0277 321401             LD     (DCONT),A     ・連続フラグのセット
027A C9                 RET

027B 77        BPR0:    LD     (HL),A
027C 23                 INC    HL
027D 3601               LD     (HL),1
027F CD7502             CALL   DCON1
0282 CD300B    BSAML:   CALL   TDATA         ・00HまたはFFHが連続した場合の処理
0285 BB                 CP     E
0286 2005               JR     NZ,TOBPR
0288 CDBE0A             CALL   INC1F
028B 18F5               JR     BSAML
028D 5F        TOBPR:   LD     E,A
028E 23                 INC    HL
028F AF                 XOR    A
0290 321401             LD     (DCONT),A     ・圧縮処理の終了
0293 47                 LD     B,A
0294 57                 LD     D,A
0295 7B                 LD     A,E
0296 C33A02             JP     BPRE4

0299 AF        BPR1:    XOR    A             ・55HまたはAAHが連続した場合の処理
029A 1802               JR     BPR2+2
029C 3E0F      BPR2:    LD     A,0FH
029E 32C202             LD     (BSPO),A
02A1 73                 LD     (HL),E
02A2 23                 INC    HL
02A3 3601               LD     (HL),1
02A5 53                 LD     D,E
02A6 CD2B0B             CALL   TDAT1
02A9 5F                 LD     E,A
02AA 3818               JR     C,BSPO+2      ・上位／下位が同じデータが連続した場合の処理
02AC BA                 CP     D
02AD 2013               JR     NZ,BSPO
02AF 3682               LD     (HL),82H
02B1 CD300B    BSPL1:   CALL   TDATA
02B4 BA                 CP     D
02B5 20D6               JR     NZ,TOBPR
02B7 CDCC0A             CALL   INC8F
02BA 18F5               JR     BSPL1

02BC CD300B    BSPL2:   CALL   TDATA
02BF 5F                 LD     E,A
02C0 3802               JR     C,BSPO+2
02C2 0F        BSPO:    RRCA
02C3 0F                 RRCA
02C4 BA                 CP     D             ・圧縮が次の縦列になった場合の処理
02C5 20C7               JR     NZ,TOBPR+1
02C7 53                 LD     D,E
02C8 CDC50A             CALL   INC17
02CB 18EF               JR     BSPL2
```

```
02CD CD300B   RPRES:  CALL  TDATA       ・データ（R面）をリードし同一縦列内なら
02D0 D41807           CALL  NC,ROTER      Cレジスタをローテート
02D3 5F               LD    E,A
02D4 79       RPRE4:  LD    A,C         ・C＝00HならRNEWへ
02D5 B7               OR    A
02D6 2808             JR    Z,RNEW
02D8 FEFF             CP    0FFH        ・C＝FFHならCFENTへ、それ以外ならCXENTへ
02DA CA9A04           JP    Z,CFENT
02DD C3DF04           JP    CXENT

02E0 7B       RNEW:   LD    A,E         ・データがB面と同じならBSET0へ
02E1 D35C             OUT   (5CH),A
02E3 D9               EXX
02E4 BE               CP    (HL)
02E5 D9               EXX
02E6 CAA604           JP    Z,BSET0
02E9 B7               OR    A           ・データ＝00HならRPR00へ
02EA CAC903           JP    Z,RPR00
02ED 3E55             LD    A,55H       ・データが「B面 AND C」ならBSETXへ
02EF 4F               LD    C,A
02F0 D9               EXX
02F1 A6               AND   (HL)
02F2 D9               EXX
02F3 BB               CP    E
02F4 CAED04           JP    Z,BSETX
02F7 3EAA             LD    A,0AAH
02F9 4F               LD    C,A
02FA D9               EXX
02FB A6               AND   (HL)
02FC D9               EXX
02FD BB               CP    E
02FE CAED04           JP    Z,BSETX
0301 3E33             LD    A,33H
0303 4F               LD    C,A
0304 D9               EXX
0305 A6               AND   (HL)
0306 D9               EXX
0307 BB               CP    E
0308 CAED04           JP    Z,BSETX
030B 3E66             LD    A,66H
030D 4F               LD    C,A
030E D9               EXX
030F A6               AND   (HL)
0310 D9               EXX
0311 BB               CP    E
0312 CAED04           JP    Z,BSETX
0315 3E99             LD    A,99H
0317 4F               LD    C,A
0318 D9               EXX
0319 A6               AND   (HL)
031A D9               EXX
031B BB               CP    E
031C CAED04           JP    Z,BSETX
031F 3ECC             LD    A,0CCH
0321 4F               LD    C,A
0322 D9               EXX
0323 A6               AND   (HL)
0324 D9               EXX
0325 BB               CP    E
0326 CAED04           JP    Z,BSETX
0329 C32C03   TYPER:  JP    TYPRA       ・Bタイプの場合はTYPRBへ
032C 3E22     TYPRA:  LD    A,22H
032E 4F               LD    C,A
032F D9               EXX
0330 A6               AND   (HL)
0331 D9               EXX
0332 BB               CP    E
0333 CAED04           JP    Z,BSETX
0336 3E77             LD    A,77H
0338 4F               LD    C,A
```

```
0339 D9              EXX
033A A6              AND     (HL)
033B D9              EXX
033C BB              CP      E
033D CAED04          JP      Z,BSETX
0340 3E88            LD      A,88H
0342 4F              LD      C,A
0343 D9              EXX
0344 A6              AND     (HL)
0345 D9              EXX
0346 BB              CP      E
0347 CAED04          JP      Z,BSETX
034A 3EDD            LD      A,0DDH
034C 4F              LD      C,A
034D D9              EXX
034E A6              AND     (HL)
034F D9              EXX
0350 BB              CP      E
0351 CAED04          JP      Z,BSETX          ・データ＝B面 AND CならBSETXへ
0354 3E11            LD      A,11H
0356 4F              LD      C,A
0357 D9              EXX
0358 A6              AND     (HL)
0359 D9              EXX
035A BB              CP      E
035B CAED04          JP      Z,BSETX
035E 3E44            LD      A,44H
0360 4F              LD      C,A
0361 D9              EXX
0362 A6              AND     (HL)
0363 D9              EXX
0364 BB              CP      E
0365 CAED04          JP      Z,BSETX
0368 3EBB            LD      A,0BBH
036A 4F              LD      C,A
036B D9              EXX
036C A6              AND     (HL)
036D D9              EXX
036E BB              CP      E
036F CAED04          JP      Z,BSETX
0372 3EEE            LD      A,0EEH
0374 4F              LD      C,A
0375 D9              EXX
0376 A6              AND     (HL)
0377 D9              EXX
0378 BB              CP      E
0379 CAED04          JP      Z,BSETX

037C 0E00    TYPRB:  LD      C,0
037E D35D            OUT     (5DH),A
0380 7B              LD      A,E              ・データ＝FFHならRPRFへ
0381 FEFF            CP      0FFH
0383 CAE703          JP      Z,RPRF
0386 CD8E01          CALL    CHKUD
0389 2826            JR      Z,RPRX           ・データの上位／下位が等しければRPRXへ
038B BA      RCKE:   CP      D
038C 2009            JR      NZ,RNORM-2
038E 04              INC     B
038F 78              LD      A,B              ・上位／下位の違うデータが4ケ以上連続した場合は
0390 FE03            CP      3                 SAMSTへ
0392 2005            JR      NZ,RNORM
0394 C35702          JP      SAMST
0397 0600            LD      B,0
0399 73      RNORM:  LD      (HL),E
039A 23              INC     HL
039B 7C              LD      A,H
039C FE80            CP      80H              ・通常のデータ処理（アドレスが8000Hを越えたら
039E D26D0B          JP      NC,MOVER          MOVERへ）
03A1 53              LD      D,E
03A2 C3CD02          JP      RPRES
```

```
03A5 5F         TORPR:   LD     E,A
03A6 23                  INC    HL
03A7 AF         TORPR2:  XOR    A
03A8 321401              LD     (DCONT),A
03AB 47                  LD     B,A
03AC 4F                  LD     C,A
03AD 57                  LD     D,A
03AE C3D402              JP     RPRE4
```

・圧縮ルールからデータが逸脱した場合の処理

```
03B1 73         RPRX:    LD     (HL),E
03B2 23                  INC    HL
03B3 3601                LD     (HL),1
03B5 53                  LD     D,E
03B6 4B                  LD     C,E
03B7 CD2B0B              CALL   TDAT1
03BA D41807              CALL   NC,ROTER
03BD 5F                  LD     E,A
03BE B9                  CP     C
03BF CA2604              JP     Z,RXSSR1
03C2 BA                  CP     D
03C3 CA1104              JP     Z,RXSS01
03C6 C3A603              JP     TORPR+1
```

・上位／下位が同じデータの２回目をチェック

```
03C9 D35D       RPR00:   OUT    (5DH),A
03CB 77                  LD     (HL),A
03CC 23                  INC    HL
03CD 3601                LD     (HL),1
03CF CD2B0B              CALL   TDAT1
03D2 B7                  OR     A
03D3 C2A503              JP     NZ,TORPR
03D6 CD7502     RPR01:   CALL   DCON1
03D9 3602                LD     (HL),2
03DB CD300B     RXX0L:   CALL   TDATA
03DE B7                  OR     A
03DF 20C4                JR     NZ,TORPR
03E1 CDC50A              CALL   INC17
03E4 C3DB03              JP     RXX0L
```

・データ＝00Hの圧縮処理

```
03E7 77         RPRF:    LD     (HL),A
03E8 23                  INC    HL
03E9 3601                LD     (HL),1
03EB CD2B0B              CALL   TDAT1
03EE FEFF                CP     0FFH
03F0 C2A503              JP     NZ,TORPR
03F3 CD7502     RPRF1:   CALL   DCON1
03F6 3602                LD     (HL),2
03F8 CD300B     RXXFL:   CALL   TDATA
03FB FEFF                CP     0FFH
03FD 20A6                JR     NZ,TORPR
03FF CDBE0A              CALL   INC1F
0402 C3F803              JP     RXXFL
```

・データ＝FFHの圧縮処理

```
0405 2B         RXSS0:   DEC    HL
0406 72                  LD     (HL),D
0407 23                  INC    HL
0408 7A                  LD     A,D
0409 3C                  INC    A
040A CAF303              JP     Z,RPRF1
040D 3D                  DEC    A
040E CAD603              JP     Z,RPR01
0411 CD7502     RXSS01:  CALL   DCON1
0414 3682                LD     (HL),82H
0416 CD300B     RXS0L:   CALL   TDATA
0419 BA                  CP     D
041A C2A503              JP     NZ,TORPR
041D CDE10A              CALL   INC8A
0420 C31604              JP     RXS0L
```

・上位／下位が同じデータの圧縮処理（データはローテートしない）

```
0423 2B         RXSSR:   DEC    HL
0424 72                  LD     (HL),D
```

```
0425 23                 INC     HL
0426 CD7502   RXSSR1:   CALL    DCON1
0429 3602               LD      (HL),2
042B 4B                 LD      C,E
042C CD300B   RXSRL:    CALL    TDATA
042F D41807             CALL    NC,ROTER
0432 B9                 CP      C
0433 C2A503             JP      NZ,TORPR
0436 CDC50A             CALL    INC17
0439 C32C04             JP      RXSRL
```

・上位／下位が同じデータの圧縮処理（データはローテートする）

```
043C 2B       BNXX0:    DEC     HL
043D 3600     EQB0:     LD      (HL),0
043F 23                 INC     HL
0440 36B2               LD      (HL),0B2H
0442 CD7502             CALL    DCON1
0445 CD300B   BXLP0:    CALL    TDATA
0448 5F                 LD      E,A
0449 79                 LD      A,C
044A D9                 EXX
044B D35C               OUT     (5CH),A
044D A6                 AND     (HL)
044E D35D               OUT     (5DH),A
0450 D9                 EXX
0451 BB                 CP      E
0452 C2A603             JP      NZ,TORPR+1
0455 CDF30A             CALL    INCBB
0458 C34504             JP      BXLP0
```

・データが「B面 AND C」の場合の圧縮処理（Cはローテートしない）

```
045B 2B       BNXXR:    DEC     HL
045C 3600     EQB1:     LD      (HL),0
045E 23                 INC     HL
045F 36C2               LD      (HL),0C2H
0461 CD7502             CALL    DCON1
0464 CD300B   BXLPR:    CALL    TDATA
0467 D41807             CALL    NC,ROTER
046A 5F                 LD      E,A
046B 79                 LD      A,C
046C D9                 EXX
046D D35C               OUT     (5CH),A
046F A6                 AND     (HL)
0470 D35D               OUT     (5DH),A
0472 D9                 EXX
0473 BB                 CP      E
0474 C2A603             JP      NZ,TORPR+1
0477 CDDD0A             CALL    INCCF
047A C36404             JP      BXLPR
```

・データが「B面 AND C」の場合の圧縮処理（Cはローテートする）

```
047D 2B       BSAME:    DEC     HL
047E 3600               LD      (HL),0
0480 23                 INC     HL
0481 3682               LD      (HL),82H
0483 CD7502             CALL    DCON1
0486 CD300B   BSMLP:    CALL    TDATA
0489 5F                 LD      E,A
048A D9                 EXX
048B D35C               OUT     (5CH),A
048D BE                 CP      (HL)
048E D35D               OUT     (5DH),A
0490 D9                 EXX
0491 C2A603             JP      NZ,TORPR+1
0494 CDCC0A             CALL    INC8F
0497 C38604             JP      BSMLP
```

・データがB面と同じ場合の圧縮処理

```
049A E5       CFENT:    PUSH    HL
049B 21E002             LD      HL,RNEW
049E 22CD04             LD      (BFFPO+1),HL
04A1 E1                 POP     HL
04A2 7B                 LD      A,E
04A3 C3C204             JP      CFENT0
```

・B面と同じで上位／下位が違う場合の２回目のチェック

```
04A6 0EFF    BSET0:  LD    C,0FFH
04A8 D35D            OUT   (5DH),A          ・B面と同じで上位／下位が違う場合はRCKEへ
04AA CD8901          CALL  ABCHK
04AD C28B03          JP    NZ,RCKE
04B0 E5              PUSH  HL
04B1 21CF04          LD    HL,DATFF
04B4 22CD04          LD    (BFFPO+1),HL
04B7 E1              POP   HL
04B8 53              LD    D,E
04B9 3600            LD    (HL),0
04BB 23              INC   HL
04BC 3681            LD    (HL),81H         ・2回目のデータがB面と同じかどうかをチェック
04BE CD2B0B          CALL  TDAT1
04C1 5F              LD    E,A
04C2 D9      CFENT0: EXX
04C3 D35C            OUT   (5CH),A
04C5 BE              CP    (HL)
04C6 D35D            OUT   (5DH),A
04C8 D9              EXX
04C9 CA7D04          JP    Z,BSAME
04CC C3CF04  BFFPO:  JP    DATFF            ・CFENTからの場合はRNEWへ
04CF BA      DATFF:  CP    D
04D0 CA0504          JP    Z,RXSS0
04D3 4A              LD    C,D
04D4 CD1807          CALL  ROTER
04D7 B9              CP    C                ・2回目のデータが前回（ローテート含む）と同じか
04D8 CA2304          JP    Z,RXSSR            どうかをチェック
04DB 23              INC   HL
04DC C3A703          JP    TORPR2

04DF AF      CXENT:  XOR   A
04E0 324105          LD    (TRPO1),A
04E3 E5              PUSH  HL
04E4 21E002          LD    HL,RNEW          ・「B面 AND C」と同じで上位／下位が違う場合の
04E7 223205          LD    (BXXPO+1),HL       2回目をチェック
04EA E1              POP   HL
04EB 1829            JR    CXENT0
04ED 79      BSETX:  LD    A,C
04EE 323E04          LD    (EQB0+1),A
04F1 325D04          LD    (EQB1+1),A
04F4 D35D            OUT   (5DH),A          ・「B面 AND C」と同じで上位／下位が違う場合は
04F6 7B              LD    A,E                RCKEへ
04F7 CD8901          CALL  ABCHK
04FA C28B03          JP    NZ,RCKE
04FD E5              PUSH  HL
04FE 213405          LD    HL,DATXX
0501 223205          LD    (BXXPO+1),HL
0504 E1              POP   HL
0505 3E23            LD    A,23H
0507 324105          LD    (TRPO1),A
050A 53              LD    D,E
050B 71              LD    (HL),C
050C 23              INC   HL
050D 36C1            LD    (HL),0C1H
050F CD2B0B          CALL  TDAT1
0512 D41807          CALL  NC,ROTER
0515 5F              LD    E,A
0516 79      CXENT0: LD    A,C
0517 D9              EXX                    ・2回目のデータが「B面 AND C」（ローテート含む）
0518 D35C            OUT   (5CH),A            と同じかどうかをチェック
051A A6              AND   (HL)
051B D35D            OUT   (5DH),A
051D D9              EXX
051E BB              CP    E
051F CA5B04          JP    Z,BNXXR
0522 CD1807          CALL  ROTER
0525 79              LD    A,C
0526 D9              EXX
0527 D35C            OUT   (5CH),A
0529 A6              AND   (HL)
```

```
052A D35D               OUT    (5DH),A
052C D9                 EXX
052D BB                 CP     E
052E CA3C04             JP     Z,BNXX0
0531 C33405    BXXPO:   JP     DATXX          ・CXENTからの場合はRNEWへ
0534 7B        DATXX:   LD     A,E
0535 BA                 CP     D
0536 CA0504             JP     Z,RXSS0
0539 4A                 LD     C,D
053A CD1807             CALL   ROTER          ・2回目のデータが前回（ローテート含む）と同じか
053D B9                 CP     C                どうかをチェック
053E CA2304             JP     Z,RXSSR
0541 23        TRPO1:   INC    HL
0542 C3A703             JP     TORPR2

0545 79        GPRES:   LD     A,C
0546 325A08             LD     (EQG0+1),A
0549 326008             LD     (BR0CK+1),A
054C 328508             LD     (EQG1+1),A
054F 328B08             LD     (RR0CK+1),A    ・プログラム上のワークエリアを初期化
0552 32B908             LD     (EQG2+1),A
0555 32C008             LD     (EQG3+1),A
0558 32D208             LD     (EQG4+1),A
055B 32D808             LD     (XX0CK+1),A
055E CD300B             CALL   TDATA          ・データ（G面）をリードし同一縦列内ならCレジス
0561 D41807             CALL   NC,ROTER         タをローテート
0564 5F                 LD     E,A
0565 79        GPRE4:   LD     A,C
0566 3C                 INC    A              ・C＝FFHならGPCFFへ
0567 2810               JR     Z,GPCFF
0569 3D                 DEC    A              ・C＝00HならGNEWへ
056A 281B               JR     Z,GNEW
056C CB78               BIT    7,B            ・Bのビット7＝1ならBRSCKへ
056E C29E08             JP     NZ,BRSCK
0571 CB70               BIT    6,B            ・Bのビット6＝1ならBSAMへ
0573 C24808             JP     NZ,BSAM
0576 C37308             JP     RSAM           ・Bのビット5＝1ならRSAMへ
0579 7B        GPCFF:   LD     A,E
057A CB78               BIT    7,B            ・Bのビット7＝1ならCFFBRへ
057C C2220A             JP     NZ,CFFBR
057F CB70               BIT    6,B            ・Bのビット6＝1ならCFFBへ
0581 C2820A             JP     NZ,CFFB
0584 C3E609             JP     CFFR           ・Bのビット5＝1ならCFFRへ

0587 0EFF      GNEW:    LD     C,0FFH
0589 7B                 LD     A,E
058A D9                 EXX
058B D35C               OUT    (5CH),A
058D BE                 CP     (HL)
058E CA400A             JP     Z,CKBRF
0591 D35D               OUT    (5DH),A
0593 BE                 CP     (HL)
0594 D9                 EXX
0595 CAFB09             JP     Z,RSETF
0598 D35C               OUT    (5CH),A
059A 3E55               LD     A,55H
059C 4F                 LD     C,A
059D D9                 EXX
059E A6                 AND    (HL)
059F D9                 EXX
05A0 BB                 CP     E
05A1 CA7409             JP     Z,CKBRX
05A4 3EAA               LD     A,0AAH
05A6 4F                 LD     C,A
05A7 D9                 EXX
05A8 A6                 AND    (HL)
05A9 D9                 EXX
05AA BB                 CP     E
05AB CA7409             JP     Z,CKBRX
05AE D35E               OUT    (5EH),A
```

```
05B0 7B          LD     A,E
05B1 B7          OR     A
05B2 CAFA06      JP     Z,GDA0F
05B5 D35D        OUT    (5DH),A
05B7 3E55        LD     A,55H
05B9 4F          LD     C,A
05BA D9          EXX
05BB A6          AND    (HL)
05BC D9          EXX
05BD BB          CP     E
05BE CA4009      JP     Z,RSETX
05C1 3EAA        LD     A,0AAH
05C3 4F          LD     C,A
05C4 D9          EXX
05C5 A6          AND    (HL)
05C6 D9          EXX
05C7 BB          CP     E
05C8 CA4009      JP     Z,RSETX

05CB D35C        OUT    (5CH),A
05CD 3E33        LD     A,33H
05CF 4F          LD     C,A
05D0 D9          EXX
05D1 A6          AND    (HL)
05D2 D9          EXX
05D3 BB          CP     E
05D4 CA7409      JP     Z,CKBRX
05D7 3E66        LD     A,66H
05D9 4F          LD     C,A
05DA D9          EXX
05DB A6          AND    (HL)
05DC D9          EXX
05DD BB          CP     E
05DE CA7409      JP     Z,CKBRX
05E1 3E99        LD     A,99H       ・データ＝00HまたはB／R面と共通（ANDも含む）
05E3 4F          LD     C,A          ならそれぞれのルーチンへ
05E4 D9          EXX
05E5 A6          AND    (HL)
05E6 D9          EXX
05E7 BB          CP     E
05E8 CA7409      JP     Z,CKBRX
05EB 3ECC        LD     A,0CCH
05ED 4F          LD     C,A
05EE D9          EXX
05EF A6          AND    (HL)
05F0 D9          EXX
05F1 BB          CP     E
05F2 CA7409      JP     Z,CKBRX
05F5 D35D        OUT    (5DH),A
05F7 3E33        LD     A,33H
05F9 4F          LD     C,A
05FA D9          EXX
05FB A6          AND    (HL)
05FC D9          EXX
05FD BB          CP     E
05FE CA4009      JP     Z,RSETX
0601 3E66        LD     A,66H
0603 4F          LD     C,A
0604 D9          EXX
0605 A6          AND    (HL)
0606 D9          EXX
0607 BB          CP     E
0608 CA4009      JP     Z,RSETX
060B 3E99        LD     A,99H
060D 4F          LD     C,A
060E D9          EXX
060F A6          AND    (HL)
0610 D9          EXX
0611 BB          CP     E
0612 CA4009      JP     Z,RSETX
```

```
0615 3ECC                 LD      A,0CCH
0617 4F                   LD      C,A
0618 D9                   EXX
0619 A6                   AND     (HL)
061A D9                   EXX
061B BB                   CP      E
061C CA4009               JP      Z,RSETX

061F C32206     TYPEG:    JP      TYPGA          ・Bタイプの場合はTYPGBへ
0622 3E22       TYPGA:    LD      A,22H
0624 D35C                 OUT     (5CH),A
0626 4F                   LD      C,A
0627 D9                   EXX
0628 A6                   AND     (HL)
0629 D9                   EXX
062A BB                   CP      E
062B CA7409               JP      Z,CKBRX
062E 3E77                 LD      A,77H
0630 4F                   LD      C,A
0631 D9                   EXX
0632 A6                   AND     (HL)
0633 D9                   EXX
0634 BB                   CP      E
0635 CA7409               JP      Z,CKBRX
0638 3E88                 LD      A,88H
063A 4F                   LD      C,A
063B D9                   EXX
063C A6                   AND     (HL)
063D D9                   EXX
063E BB                   CP      E
063F CA7409               JP      Z,CKBRX
0642 3EDD                 LD      A,0DDH
0644 4F                   LD      C,A
0645 D9                   EXX
0646 A6                   AND     (HL)
0647 D9                   EXX
0648 BB                   CP      E
0649 CA7409               JP      Z,CKBRX

064C D35D                 OUT     (5DH),A
064E 3E22                 LD      A,22H
0650 4F                   LD      C,A
0651 D9                   EXX
0652 A6                   AND     (HL)
0653 D9                   EXX
0654 BB                   CP      E
0655 CA4009               JP      Z,RSETX
0658 3E77                 LD      A,77H
065A 4F                   LD      C,A
065B D9                   EXX
065C A6                   AND     (HL)
065D D9                   EXX
065E BB                   CP      E
065F CA4009               JP      Z,RSETX
0662 3E88                 LD      A,88H
0664 4F                   LD      C,A
0665 D9                   EXX
0666 A6                   AND     (HL)
0667 D9                   EXX
0668 BB                   CP      E
0669 CA4009               JP      Z,RSETX
066C 3EDD                 LD      A,0DDH
066E 4F                   LD      C,A
066F D9                   EXX
0670 A6                   AND     (HL)           ・B／R面と共通（ANDも含む）ならそれぞれの
0671 D9                   EXX                     ルーチンへ
0672 BB                   CP      E
0673 CA4009               JP      Z,RSETX

0676 D35C                 OUT     (5CH),A
```

```
0678 3E11                LD      A,11H
067A 4F                  LD      C,A
067B D9                  EXX
067C A6                  AND     (HL)
067D D9                  EXX
067E BB                  CP      E
067F CA7409              JP      Z,CKBRX
0682 3E44                LD      A,44H
0684 4F                  LD      C,A
0685 D9                  EXX
0686 A6                  AND     (HL)
0687 D9                  EXX
0688 BB                  CP      E
0689 CA7409              JP      Z,CKBRX
068C 3EBB                LD      A,0BBH
068E 4F                  LD      C,A
068F D9                  EXX
0690 A6                  AND     (HL)
0691 D9                  EXX
0692 BB                  CP      E
0693 CA7409              JP      Z,CKBRX
0696 3EEE                LD      A,0EEH
0698 4F                  LD      C,A
0699 D9                  EXX
069A A6                  AND     (HL)
069B D9                  EXX
069C BB                  CP      E
069D CA7409              JP      Z,CKBRX

06A0 D35D                OUT     (5DH),A
06A2 3E11                LD      A,11H
06A4 4F                  LD      C,A
06A5 D9                  EXX
06A6 A6                  AND     (HL)
06A7 D9                  EXX
06A8 BB                  CP      E
06A9 CA4009              JP      Z,RSETX
06AC 3E44                LD      A,44H
06AE 4F                  LD      C,A
06AF D9                  EXX
06B0 A6                  AND     (HL)
06B1 D9                  EXX
06B2 BB                  CP      E
06B3 CA4009              JP      Z,RSETX
06B6 3EBB                LD      A,0BBH
06B8 4F                  LD      C,A
06B9 D9                  EXX
06BA A6                  AND     (HL)
06BB D9                  EXX
06BC BB                  CP      E
06BD CA4009              JP      Z,RSETX
06C0 3EEE                LD      A,0EEH
06C2 4F                  LD      C,A
06C3 D9                  EXX
06C4 A6                  AND     (HL)
06C5 D9                  EXX
06C6 BB                  CP      E
06C7 CA4009              JP      Z,RSETX

06CA 78         TYPGB:   LD      A,B
06CB 0E00                LD      C,0
06CD D35E                OUT     (5EH),A
06CF E60F                AND     0FH
06D1 47                  LD      B,A
06D2 7B                  LD      A,E
06D3 FEFF                CP      0FFH
06D5 2823                JR      Z,GDA0F
06D7 CD8E01              CALL    CHKUD
06DA 284C                JR      Z,GDAXX
06DC BA         GCKE:    CP      D
```

・データ＝FFHまたは上位／下位が等しければそれぞれのルーチンへ

```
06DD 200B                JR     NZ,GNOR0
06DF 04                  INC    B
06E0 78                  LD     A,B
06E1 E60F                AND    0FH             ・同じデータが4ケ以上連続した場合はSAMSTへ
06E3 FE03                CP     3
06E5 2007                JR     NZ,GNORM
06E7 C35702              JP     SAMST
06EA CB80      GNOR0:    RES    0,B
06EC CB88                RES    1,B
06EE 73        GNORM:    LD     (HL),E
06EF 23                  INC    HL
06F0 7C                  LD     A,H             ・通常のデータ処理（アドレスが8000Hを越えたら
06F1 FE80                CP     80H               MOVERへ）
06F3 D26D0B              JP     NC,MOVER
06F6 53                  LD     D,E
06F7 C34505              JP     GPRES

06FA 77        GDA0F:    LD     (HL),A
06FB 23                  INC    HL
06FC 3601                LD     (HL),1
06FE CD7502              CALL   DCON1
0701 CD300B    GSAML:    CALL   TDATA           ・データ＝00HまたはFFHの場合の圧縮処理
0704 BB                  CP     E
0705 2005                JR     NZ,TOGPR
0707 CDC50A    GUP0F:    CALL   INC17
070A 18F5                JR     GSAML
070C 5F        TOGPR:    LD     E,A
070D 23                  INC    HL
070E AF                  XOR    A
070F 321401              LD     (DCONT),A       ・圧縮ルールからデータが逸脱した場合の処理
0712 47                  LD     B,A
0713 4F                  LD     C,A
0714 57                  LD     D,A
0715 C36505              JP     GPRE4

0718 F5        ROTER:    PUSH   AF
0719 79                  LD     A,C
071A FE55                CP     55H
071C 2806                JR     Z,ROT11
071E FEAA                CP     0AAH            ・Cレジスタの値をローテートする
0720 2802                JR     Z,ROT11
0722 CB09                RRC    C
0724 CB09      ROT11:    RRC    C
0726 F1                  POP    AF
0727 C9                  RET

0728 73        GDAXX:    LD     (HL),E
0729 23                  INC    HL
072A 3601                LD     (HL),1
072C 53                  LD     D,E
072D 4B                  LD     C,E
072E CD2B0B              CALL   TDAT1           ・2回目のデータチェック
0731 D41807              CALL   NC,ROTER          1回目＝上位／下位が同じ
0734 5F                  LD     E,A
0735 B9                  CP     C
0736 2014                JR     NZ,GSACK
0738 3602      SROTX2:   LD     (HL),2
073A CD7502              CALL   DCON1
073D CD300B    ROTXX:    CALL   TDATA
0740 D41807              CALL   NC,ROTER
0743 5F                  LD     E,A             ・上位／下位が同じデータの圧縮処理（データは
0744 B9                  CP     C                 ローテートする）
0745 20C6                JR     NZ,TOGPR+1
0747 CDD50A              CALL   INC13
074A 18F1                JR     ROTXX

074C BA        GSACK:    CP     D
074D C20D07              JP     NZ,TOGPR+1
0750 180C                JR     SSAMX2
0752 CD7502    SAMFX:    CALL   DCON1
```

```
0755 7B                  LD      A,E
0756 3C                  INC     A
0757 CA0707              JP      Z,GUP0F
075A 3D                  DEC     A
075B CA0707              JP      Z,GUP0F
075E 3642      SSAMX2:   LD      (HL),42H       ・上位／下位が同じデータの圧縮処理（データは
0760 CD7502              CALL    DCON1            ローテートしない）
0763 CD300B    SAMXX:    CALL    TDATA
0766 BA                  CP      D
0767 C20C07              JP      NZ,TOGPR
076A CDEA0A              CALL    INC45
076D 18F4                JR      SAMXX

076F 3662      SBXX0:    LD      (HL),62H
0771 CD7502              CALL    DCON1
0774 CD300B    BXXL0:    CALL    TDATA
0777 5F                  LD      E,A
0778 79                  LD      A,C
0779 D9                  EXX
077A D35C                OUT     (5CH),A        ・データが「B面 AND C」の場合の圧縮処理
077C A6                  AND     (HL)             （Cはローテートしない）
077D D35E                OUT     (5EH),A
077F D9                  EXX
0780 BB                  CP      E
0781 C20D07              JP      NZ,TOGPR+1
0784 CDF70A    BSSTX:    CALL    INC66
0787 18EB                JR      BXXL0

0789 3672      SRXX0:    LD      (HL),72H
078B CD7502              CALL    DCON1
078E CD300B    RXXL0:    CALL    TDATA
0791 5F                  LD      E,A
0792 79        CRAND0:   LD      A,C
0793 D9                  EXX
0794 D35D                OUT     (5DH),A        ・データが「R面 AND C」の場合の圧縮処理
0796 A6                  AND     (HL)             （Cはローテートしない）
0797 D35E                OUT     (5EH),A
0799 D9                  EXX
079A BB                  CP      E
079B C20D07              JP      NZ,TOGPR+1
079E CDFB0A              CALL    INC77
07A1 18EB                JR      RXXL0

07A3 3682      SBXXR:    LD      (HL),82H
07A5 CD7502              CALL    DCON1
07A8 CD300B    BXXLR:    CALL    TDATA
07AB D41807              CALL    NC,ROTER
07AE 5F                  LD      E,A
07AF 79                  LD      A,C
07B0 D9                  EXX
07B1 D35C                OUT     (5CH),A        ・データが「B面 AND C」の場合の圧縮処理
07B3 A6                  AND     (HL)             （Cはローテートする）
07B4 D35E                OUT     (5EH),A
07B6 D9                  EXX
07B7 BB                  CP      E
07B8 C20D07              JP      NZ,TOGPR+1
07BB CDD90A    BRSTX:    CALL    INC8B
07BE 18E8                JR      BXXLR

07C0 36C2      SRXXR:    LD      (HL),0C2H
07C2 CD7502              CALL    DCON1
07C5 CD300B    RXXLR:    CALL    TDATA
07C8 D41807              CALL    NC,ROTER
07CB 5F                  LD      E,A
07CC 79        CRANDR:   LD      A,C
07CD D9                  EXX
07CE D35D                OUT     (5DH),A        ・データが「R面 AND C」の場合の圧縮処理
07D0 A6                  AND     (HL)             （Cはローテートする）
07D1 D35E                OUT     (5EH),A
07D3 D9                  EXX
```

```
07D4 BB                 CP    E
07D5 C20D07             JP    NZ,TOGPR+1
07D8 CDDD0A             CALL  INCCF
07DB 18E8               JR    RXXLR

07DD 3662      WWXX0:   LD    (HL),62H
07DF CD7502             CALL  DCON1
07E2 CD300B    WXXL0:   CALL  TDATA
07E5 5F                 LD    E,A
07E6 79                 LD    A,C
07E7 D9                 EXX
07E8 D35C               OUT   (5CH),A
07EA A6                 AND   (HL)
07EB D35E               OUT   (5EH),A
07ED D9                 EXX
07EE BB                 CP    E
07EF 2011               JR    NZ,TRCON
07F1 79                 LD    A,C
07F2 D9                 EXX
07F3 D35D               OUT   (5DH),A
07F5 A6                 AND   (HL)
07F6 D35E               OUT   (5EH),A
07F8 D9                 EXX
07F9 BB                 CP    E
07FA C28407             JP    NZ,BSSTX
07FD CDF70A             CALL  INC66
0800 18E0               JR    WXXL0
0802 E5        TRCON:   PUSH  HL
0803 3A1401             LD    A,(DCONT)
0806 3D                 DEC   A
0807 2802               JR    Z,TRDC0
0809 2B                 DEC   HL
080A 2B                 DEC   HL
080B CBE6      TRDC0:   SET   4,(HL)
080D E1                 POP   HL
080E C39207             JP    CRAND0

0811 3682      WWXXR:   LD    (HL),82H
0813 CD7502             CALL  DCON1
0816 CD300B    WXXLR:   CALL  TDATA
0819 D41807             CALL  NC,ROTER
081C 5F                 LD    E,A
081D 79                 LD    A,C
081E D9                 EXX
081F D35C               OUT   (5CH),A
0821 A6                 AND   (HL)
0822 D35E               OUT   (5EH),A
0824 D9                 EXX
0825 BB                 CP    E
0826 2011               JR    NZ,TRCON2
0828 79                 LD    A,C
0829 D9                 EXX
082A D35D               OUT   (5DH),A
082C A6                 AND   (HL)
082D D35E               OUT   (5EH),A
082F D9                 EXX
0830 BB                 CP    E
0831 C2BB07             JP    NZ,BRSTX
0834 CDD90A             CALL  INC8B
0837 18DD               JR    WXXLR
0839 E5        TRCON2:  PUSH  HL
083A 3A1401             LD    A,(DCONT)
083D 3D                 DEC   A
083E 2802               JR    Z,TRDC00
0840 2B                 DEC   HL
0841 2B                 DEC   HL
0842 CBF6      TRDC00:  SET   6,(HL)
0844 E1                 POP   HL
0845 C3CC07             JP    CRANDR
```

・データが「B面 AND C」「R面 AND C」と同じ場合（Cはローテートしない）長く連続するほうを求める

・データが「B面 AND C」「R面 AND C」と同じ場合（Cはローテートする）長く連続するほうを求める

```
0848 78         BSAM:   LD      A,B
0849 E60F               AND     0FH
084B 47                 LD      B,A
084C 79                 LD      A,C
084D D9                 EXX
084E D35C               OUT     (5CH),A
0850 A6                 AND     (HL)
0851 D35E               OUT     (5EH),A       ・2回目のデータチェック（Cはローテートする）
0853 D9                 EXX                    1回目＝「B面 AND C」と同じ，上位／下位は違う
0854 BB                 CP      E
0855 C25F08             JP      NZ,BR0CK
0858 2B                 DEC     HL
0859 3600       EQG0:   LD      (HL),0
085B 23                 INC     HL
085C C3A307             JP      SBXXR

085F 0E00       BR0CK:  LD      C,0
0861 79                 LD      A,C
0862 D9                 EXX
0863 D35C               OUT     (5CH),A
0865 A6                 AND     (HL)
0866 D9                 EXX
0867 D35E               OUT     (5EH),A       ・2回目のデータチェック（Cはローテートしない）
0869 BB                 CP      E              1回目＝「B面 AND C」と同じ，上位／下位は違う
086A C28705             JP      NZ,GNEW
086D 2B                 DEC     HL
086E 71                 LD      (HL),C
086F 23                 INC     HL
0870 C36F07             JP      SBXX0

0873 78         RSAM:   LD      A,B
0874 E60F               AND     0FH
0876 47                 LD      B,A
0877 79                 LD      A,C
0878 D9                 EXX
0879 D35D               OUT     (5DH),A
087B A6                 AND     (HL)
087C D35E               OUT     (5EH),A       ・2回目のデータチェック（Cはローテートする）
087E D9                 EXX                    1回目＝「R面 AND C」と同じ，上位／下位は違う
087F BB                 CP      E
0880 C28A08             JP      NZ,RR0CK
0883 2B                 DEC     HL
0884 3600       EQG1:   LD      (HL),0
0886 23                 INC     HL
0887 C3C007             JP      SRXXR

088A 0E00       RR0CK:  LD      C,0
088C 79                 LD      A,C
088D D9                 EXX
088E D35D               OUT     (5DH),A
0890 A6                 AND     (HL)
0891 D35E               OUT     (5EH),A
0893 D9                 EXX                   ・2回目のデータチェック（Cはローテートしない）
0894 BB                 CP      E              1回目＝「R面 AND C」と同じ，上位／下位は違う
0895 C28705             JP      NZ,GNEW
0898 2B                 DEC     HL
0899 71                 LD      (HL),C
089A 23                 INC     HL
089B C38907             JP      SRXX0

089E 78         BRSCK:  LD      A,B
089F E60F               AND     0FH
08A1 47                 LD      B,A
08A2 79                 LD      A,C
08A3 D35C               OUT     (5CH),A
08A5 D9                 EXX
08A6 A6                 AND     (HL)
08A7 D9                 EXX
08A8 BB                 CP      E
08A9 C2C508             JP      NZ,SCKRM
```

```
08AC 79                 LD     A,C
08AD D9                 EXX
08AE D35D               OUT    (5DH),A
08B0 A6                 AND    (HL)
08B1 D35E               OUT    (5EH),A
08B3 D9                 EXX
08B4 BB                 CP     E
08B5 2807               JR     Z,TWXXR
08B7 2B                 DEC    HL
08B8 3600       EQG2:   LD     (HL),0
08BA 23                 INC    HL
08BB C3A307             JP     SBXXR
08BE 2B         TWXXR:  DEC    HL
08BF 3600       EQG3:   LD     (HL),0
08C1 23                 INC    HL
08C2 C31108             JP     WWXXR
```

・2回目のデータチェック（Cはローテートする）
1回目=「B／R面 AND C」と同じ，上位／下位は違う

```
08C5 79         SCKRM:  LD     A,C
08C6 D9                 EXX
08C7 D35D               OUT    (5DH),A
08C9 A6                 AND    (HL)
08CA D35E               OUT    (5EH),A
08CC D9                 EXX
08CD BB                 CP     E
08CE 2007               JR     NZ,XX0CK
08D0 2B                 DEC    HL
08D1 3600       EQG4:   LD     (HL),0
08D3 23                 INC    HL
08D4 C3C007             JP     SRXXR
```

・2回目のデータチェック（Cはローテートする）
1回目=「B／R面 AND C」と同じ，上位／下位は違う
2回目=「B面 AND C」でない場合

```
08D7 0E00       XX0CK:  LD     C,0
08D9 79                 LD     A,C
08DA D9                 EXX
08DB D35C               OUT    (5CH),A
08DD A6                 AND    (HL)
08DE D9                 EXX
08DF BB                 CP     E
08E0 C28A08             JP     NZ,RR0CK
08E3 2B                 DEC    HL
08E4 71                 LD     (HL),C
08E5 23                 INC    HL
08E6 79                 LD     A,C
08E7 D9                 EXX
08E8 D35D               OUT    (5DH),A
08EA A6                 AND    (HL)
08EB D35E               OUT    (5EH),A
08ED D9                 EXX
08EE BB                 CP     E
08EF CADD07             JP     Z,WWXX0
08F2 C36F07             JP     SBXX0
```

・2回目のデータチェック（Cはローテートしない）
1回目=「B／R面 AND C」と同じ，上位／下位は違う

```
08F5 D35E       BSETXX: OUT    (5EH),A
08F7 7B                 LD     A,E
08F8 B7                 OR     A
08F9 CAFA06             JP     Z,GDA0F
08FC CBF0               SET    6,B
08FE CD8E01             CALL   CHKUD
0901 C2DC06             JP     NZ,GCKE
0904 53                 LD     D,E
0905 71                 LD     (HL),C
0906 23                 INC    HL
0907 3681               LD     (HL),81H
0909 CD2B0B             CALL   TDAT1
090C D41807             CALL   NC,ROTER
090F 5F                 LD     E,A
0910 79                 LD     A,C
0911 D9                 EXX
0912 D35C               OUT    (5CH),A
0914 A6                 AND    (HL)
0915 D35E               OUT    (5EH),A
```

・データ＝00Hの場合はGDA0Fへ

・上位≠下位の場合はGCKEへ

```
0917 D9                 EXX
0918 BB                 CP      E
0919 CAA307             JP      Z,SBXXR
091C CD1807             CALL    ROTER
091F 79                 LD      A,C
0920 D9                 EXX
0921 D35C               OUT     (5CH),A         ・2回目のデータチェック
0923 A6                 AND     (HL)             1回目=「B面 AND C」と同じ,上位／下位は同じ
0924 D35E               OUT     (5EH),A
0926 D9                 EXX
0927 BB                 CP      E
0928 CA6F07             JP      Z,SBXX0
092B 2B         CONCK:  DEC     HL
092C 72                 LD      (HL),D
092D 23                 INC     HL
092E 3601               LD      (HL),1
0930 7B                 LD      A,E
0931 BA                 CP      D
0932 CA5207             JP      Z,SAMFX
0935 CD1807             CALL    ROTER
0938 79                 LD      A,C
0939 BB                 CP      E
093A CA3807             JP      Z,SROTX2
093D C30D07             JP      TOGPR+1

0940 CBE8       RSETX:  SET     5,B
0942 D35E               OUT     (5EH),A         ・上位≠下位の場合はGCKEへ
0944 CD8901             CALL    ABCHK
0947 C2DC06             JP      NZ,GCKE
094A 53                 LD      D,E
094B 71                 LD      (HL),C
094C 23                 INC     HL
094D 36C1               LD      (HL),0C1H
094F CD2B0B             CALL    TDAT1
0952 D41807             CALL    NC,ROTER
0955 5F                 LD      E,A
0956 79                 LD      A,C
0957 D9                 EXX
0958 D35D               OUT     (5DH),A
095A A6                 AND     (HL)
095B D35E               OUT     (5EH),A
095D D9                 EXX                     ・2回目のデータチェック
095E BB                 CP      E                1回目=「R面 AND C」と同じ,上位／下位は同じ
095F CAC007             JP      Z,SRXXR
0962 CD1807             CALL    ROTER
0965 79                 LD      A,C
0966 D9                 EXX
0967 D35D               OUT     (5DH),A
0969 A6                 AND     (HL)
096A D35E               OUT     (5EH),A
096C D9                 EXX
096D BB                 CP      E
096E CA8907             JP      Z,SRXX0
0971 C32B09             JP      CONCK

0974 79         CKBRX:  LD      A,C
0975 D9                 EXX
0976 D35D               OUT     (5DH),A
0978 A6                 AND     (HL)
0979 D9                 EXX
097A BB                 CP      E
097B C2F508             JP      NZ,BSETXX
097E CBF8               SET     7,B
0980 D35E               OUT     (5EH),A
0982 CD8901             CALL    ABCHK
0985 C2DC06             JP      NZ,GCKE
0988 53                 LD      D,E
0989 71                 LD      (HL),C
098A 23                 INC     HL
098B 3681               LD      (HL),81H
```

```
098D CD2B0B              CALL   TDAT1
0990 D41807              CALL   NC,ROTER
0993 5F                  LD     E,A
0994 79                  LD     A,C
0995 D9                  EXX
0996 D35C                OUT    (5CH),A
0998 A6                  AND    (HL)
0999 D35E                OUT    (5EH),A
099B D9                  EXX
099C BB                  CP     E
099D 200F                JR     NZ,CKRBR
099F 79                  LD     A,C
09A0 D9                  EXX
09A1 D35D                OUT    (5DH),A
09A3 A6                  AND    (HL)
09A4 D35E                OUT    (5EH),A
09A6 D9                  EXX
09A7 BB                  CP     E
09A8 C2A307              JP     NZ,SBXXR
09AB C31108              JP     WWXXR
09AE 79         CKRBR:   LD     A,C
09AF D9                  EXX                 ・2回目のデータチェック
09B0 D35D                OUT    (5DH),A        1回目＝「B／R面 AND C」と同じ，上位／下位は同
09B2 A6                  AND    (HL)           じ
09B3 D35E                OUT    (5EH),A
09B5 D9                  EXX
09B6 BB                  CP     E
09B7 CAC007              JP     Z,SRXXR
09BA CD1807              CALL   ROTER
09BD 79                  LD     A,C
09BE D9                  EXX
09BF D35C                OUT    (5CH),A
09C1 A6                  AND    (HL)
09C2 D35E                OUT    (5EH),A
09C4 D9                  EXX
09C5 BB                  CP     E
09C6 200F                JR     NZ,CKRB0
09C8 79                  LD     A,C
09C9 D9                  EXX
09CA D35D                OUT    (5DH),A
09CC A6                  AND    (HL)
09CD D35E                OUT    (5EH),A
09CF D9                  EXX
09D0 BB                  CP     E
09D1 C26F07              JP     NZ,SBXX0
09D4 C3DD07              JP     WWXX0
09D7 79         CKRB0:   LD     A,C
09D8 D9                  EXX
09D9 D35D                OUT    (5DH),A
09DB A6                  AND    (HL)
09DC D35E                OUT    (5EH),A
09DE D9                  EXX
09DF BB                  CP     E
09E0 CA8907              JP     Z,SRXX0
09E3 C32B09              JP     CONCK

09E6 CBA8       CFFR:    RES    5,B
09E8 D9                  EXX
09E9 D35D       RCF00:   OUT    (5DH),A
09EB BE                  CP     (HL)
09EC D35E                OUT    (5EH),A
09EE D9                  EXX                 ・2回目のデータチェック
09EF C28705              JP     NZ,GNEW        1回目＝R面と同じ，上位／下位は違う
09F2 2B                  DEC    HL
09F3 36FF                LD     (HL),0FFH
09F5 23                  INC    HL
09F6 3682                LD     (HL),82H
09F8 C30D0A              JP     RFST
09FB D35E       RSETF:   OUT    (5EH),A
09FD B7                  OR     A            ・データ＝00Hの場合はGDAOFへ
```

```
09FE CAFA06            JP      Z,GDA0F
0A01 CBE8              SET     5,B          ・上位≠下位の場合はGCKEへ
0A03 CD8B01            CALL    ABCK2
0A06 C2DC06            JP      NZ,GCKE
0A09 71                LD      (HL),C
0A0A 23                INC     HL
0A0B 3681              LD      (HL),81H
0A0D CD7502  RFST:     CALL    DCON1
0A10 CD300B  RFLP:     CALL    TDATA
0A13 D9      RSSTF:    EXX
0A14 D35D              OUT     (5DH),A      ・2回目以降のデータチェック
0A16 BE                CP      (HL)           1回目=「R面 AND C」と同じ，上位／下位は同じ
0A17 D35E              OUT     (5EH),A
0A19 D9                EXX
0A1A C20C07            JP      NZ,TOGPR
0A1D CDCC0A            CALL    INC8F
0A20 18EE              JR      RFLP

0A22 CBB8    CFFBR:    RES     7,B
0A24 D9                EXX
0A25 D35C              OUT     (5CH),A
0A27 BE                CP      (HL)
0A28 C2E909            JP      NZ,RCF00
0A2B D35D              OUT     (5DH),A
0A2D BE                CP      (HL)
0A2E D35E              OUT     (5EH),A      ・2回目のデータチェック
0A30 D9                EXX                    1回目=B／R面と同じ，上位／下位は違う
0A31 C28E0A            JP      NZ,STBSM
0A34 2B                DEC     HL
0A35 36FF              LD      (HL),0FFH
0A37 23                INC     HL
0A38 3682              LD      (HL),82H
0A3A CD7502            CALL    DCON1
0A3D C3580A            JP      BRFL

0A40 D35D    CKBRF:    OUT     (5DH),A
0A42 BE                CP      (HL)
0A43 D9                EXX                  ・データがB面と同じでR面と違う場合はBSTFへ
0A44 D35E              OUT     (5EH),A
0A46 C29A0A            JP      NZ,BSTF
0A49 CBF8              SET     7,B
0A4B CD8901            CALL    ABCHK        ・上位≠下位の場合はGCKEへ
0A4E C2DC06            JP      NZ,GCKE
0A51 71                LD      (HL),C
0A52 23                INC     HL
0A53 3681              LD      (HL),81H
0A55 CD7502            CALL    DCON1
0A58 CD300B  BRFL:     CALL    TDATA
0A5B D9                EXX
0A5C D35C              OUT     (5CH),A
0A5E BE                CP      (HL)
0A5F D9                EXX                  ・データがB／R面と同じ場合
0A60 C2130A            JP      NZ,RSSTF       長く連続するほうを求める
0A63 D35D              OUT     (5DH),A
0A65 D9                EXX
0A66 BE                CP      (HL)
0A67 D9                EXX
0A68 C2720A            JP      NZ,BSSTF
0A6B D35E              OUT     (5EH),A
0A6D CDCC0A            CALL    INC8F
0A70 18E6              JR      BRFL

0A72 E5      BSSTF:    PUSH    HL
0A73 5F                LD      E,A
0A74 3A1401            LD      A,(DCONT)
0A77 3D                DEC     A
0A78 2802              JR      Z,BSFD1
0A7A 2B                DEC     HL           ・データが「B／R面と同じ」から「B面と同じ」へ
0A7B 2B                DEC     HL             決定した場合の圧縮処理
0A7C 2B      BSFD1:    DEC     HL
```

```
0A7D 34                  INC     (HL)
0A7E E1                  POP     HL
0A7F 7B                  LD      A,E
0A80 182D                JR      BFENT

0A82 CBB0     CFFB:      RES     6,B
0A84 D9                  EXX
0A85 D35C                OUT     (5CH),A
0A87 BE                  CP      (HL)
0A88 D35E                OUT     (5EH),A
0A8A D9                  EXX
0A8B C28705              JP      NZ,GNEW        ・2回目のデータチェック 1回目=B面と同じ,
0A8E 2B       STBSM:     DEC     HL               上位／下位は違う
0A8F 3600                LD      (HL),0
0A91 23                  INC     HL
0A92 3682                LD      (HL),82H
0A94 CD7502              CALL    DCON1
0A97 C3AC0A              JP      BSFL

0A9A CBF0     BSTF:      SET     6,B
0A9C D35E                OUT     (5EH),A        ・データがB面と同じで上位≠下位の場合はGCKEへ
0A9E CD8901              CALL    ABCHK
0AA1 C2DC06              JP      NZ,GCKE
0AA4 3600                LD      (HL),0
0AA6 23                  INC     HL
0AA7 3681                LD      (HL),81H
0AA9 CD7502              CALL    DCON1
0AAC CD300B   BSFL:      CALL    TDATA
0AAF D9       BFENT:     EXX
0AB0 D35C                OUT     (5CH),A        ・データがB面と同じ場合の圧縮処理
0AB2 BE                  CP      (HL)
0AB3 D35E                OUT     (5EH),A
0AB5 D9                  EXX
0AB6 C20C07              JP      NZ,TOGPR
0AB9 CDCC0A              CALL    INC8F
0ABC 18EE                JR      BSFL

0ABE 3EFF     INC1F:     LD      A,0FFH
0AC0 32150B              LD      (IMAX+1),A
0AC3 1844                JR      INCM1
0AC5 3E7F     INC17:     LD      A,7FH
0AC7 32150B              LD      (IMAX+1),A
0ACA 183D                JR      INCM1
0ACC 3EFF     INC8F:     LD      A,0FFH
0ACE 32150B              LD      (IMAX+1),A
0AD1 3E7F                LD      A,7FH
0AD3 1834                JR      INCM1
0AD5 3E3F     INC13:     LD      A,3FH
0AD7 182B                JR      INCM
0AD9 3EBF     INC8B:     LD      A,0BFH
0ADB 1827                JR      INCM
0ADD 3EFF     INCCF:     LD      A,0FFH
0ADF 1823                JR      INCM
0AE1 3EAF     INC8A:     LD      A,0AFH
0AE3 32150B              LD      (IMAX+1),A
0AE6 3E2F                LD      A,2FH
0AE8 181F                JR      INCM1
0AEA 3E5F     INC45:     LD      A,5FH
0AEC 32150B              LD      (IMAX+1),A
0AEF 3E1F                LD      A,1FH
0AF1 1816                JR      INCM1
0AF3 3EBF     INCBB:     LD      A,0BFH
0AF5 1806                JR      IC0F
0AF7 3E6F     INC66:     LD      A,6FH
0AF9 1802                JR      IC0F           ・圧縮に合わせたカウント・アップ
0AFB 3E7F     INC77:     LD      A,7FH
0AFD 32150B   IC0F:      LD      (IMAX+1),A
0B00 3E0F                LD      A,0FH
0B02 1805                JR      INCM1
0B04 32150B   INCM:      LD      (IMAX+1),A
```

```
0B07 3E3F                  LD      A,3FH
0B09 32220B     INCM1:     LD      (BBSRT+1),A
0B0C 34                    INC     (HL)
0B0D 3A1401                LD      A,(DCONT)
0B10 FE02                  CP      2
0B12 2810                  JR      Z,IHLBB
0B14 3EFF       IMAX:      LD      A,0FFH
0B16 BE                    CP      (HL)
0B17 C0                    RET     NZ
0B18 3E02                  LD      A,2
0B1A 321401                LD      (DCONT),A
0B1D 23                    INC     HL
0B1E 3600                  LD      (HL),0
0B20 23                    INC     HL
0B21 3600       BBSRT:     LD      (HL),0
0B23 C9                    RET
0B24 AF         IHLBB:     XOR     A
0B25 BE                    CP      (HL)
0B26 C0                    RET     NZ
0B27 2B                    DEC     HL
0B28 34                    INC     (HL)
0B29 23                    INC     HL
0B2A C9                    RET

0B2B 3E01       TDAT1:     LD      A,1           ・圧縮ルーチン中で２回目のデータを得る
0B2D 321401                LD      (DCONT),A
0B30 D9         TDATA:     EXX
0B31 CB41                  BIT     0,C
0B33 2808                  JR      Z,ADDDE
0B35 05                    DEC     B
0B36 2811                  JR      Z,TOADD
0B38 B7                    OR      A
0B39 ED52                  SBC     HL,DE
0B3B 1804                  JR      ADDD1
0B3D 05         ADDDE:     DEC     B             ・１バイトの画面データを得る
0B3E 2809                  JR      Z,TOADD
0B40 19                    ADD     HL,DE
0B41 7E         ADDD1:     LD      A,(HL)
0B42 B7                    OR      A
0B43 D9                    EXX
0B44 C8                    RET     Z
0B45 327E0B                LD      (DTCK0+1),A
0B48 C9                    RET

0B49 0C         TOADD:     INC     C
0B4A 3A0801                LD      A,(XSIZE)
0B4D B9                    CP      C
0B4E 280F                  JR      Z,TDEND
0B50 23                    INC     HL
0B51 3A0901                LD      A,(YSIZE)
0B54 47                    LD      B,A
0B55 7E                    LD      A,(HL)        ・次の縦列に入った場合の処理（Cフラグを立てて
0B56 B7                    OR      A               戻る）
0B57 2803                  JR      Z,TOAD1
0B59 327E0B                LD      (DTCK0+1),A
0B5C D9         TOAD1:     EXX
0B5D 37                    SCF
0B5E C9                    RET

0B5F D9         TDEND:     EXX
0B60 F1                    POP     AF
0B61 3A1401                LD      A,(DCONT)
0B64 B7                    OR      A
0B65 2801                  JR      Z,TDE1
0B67 23                    INC     HL
0B68 7C         TDE1:      LD      A,H
0B69 FE80                  CP      80H
0B6B 380F                  JR      C,CKEXT
0B6D 21FC7F     MOVER:     LD      HL,7FFCH
0B70 3601                  LD      (HL),1
```

```
0B72 23                 INC    HL
0B73 3600               LD     (HL),0             ・最終データ時の処理
0B75 23                 INC    HL                 「POP AF」はSP合わせのためのダミー
0B76 36FF               LD     (HL),0FFH          圧縮データ・メモリチェック
0B78 23                 INC    HL
0B79 36FF               LD     (HL),0FFH
0B7B 23                 INC    HL
0B7C EB        CKEXT:   EX     DE,HL
0B7D 3E00      DTCK0:   LD     A,0
0B7F B7                 OR     A
0B80 2805               JR     Z,EXRET
0B82 ED531001  EXIST:   LD     (ATOP),DE
0B86 37                 SCF
0B87 D35F      EXRET:   OUT    (5FH),A
0B89 C9                 RET

0B8A DD2A0C01  OPEN3:   LD     IX,(DDTOP)
0B8E DD7E00             LD     A,(IX+0)
0B91 0F                 RRCA
0B92 21A50C             LD     HL,DTYPB1
0B95 0F                 RRCA
0B96 3003               JR     NC,DTPS1
0B98 21B40C             LD     HL,DTYPA1
0B9B 22A30C    DTPS1:   LD     (DATAT1+1),HL
0B9E 0F                 RRCA
0B9F 21380D             LD     HL,DTYPB2          ・プレーン別のデータタイプ（A／B）によりプログ
0BA2 0F                 RRCA                      ラムを書き換える
0BA3 3003               JR     NC,DTPS2
0BA5 21470D             LD     HL,DTYPA2
0BA8 22360D    DTPS2:   LD     (DATAT2+1),HL
0BAB 0F                 RRCA
0BAC 211D0E             LD     HL,DTYPB3
0BAF 0F                 RRCA
0BB0 3003               JR     NC,CKPSE
0BB2 212C0E             LD     HL,DTYPA3
0BB5 221B0E    CKPSE:   LD     (DATAT3+1),HL
0BB8 AF                 XOR    A
0BB9 32720C             LD     (SHIF1+1),A
0BBC 32F40C             LD     (SHIF2+1),A
0BBF 32EE0D             LD     (SHIF3+1),A
0BC2 2A0E01             LD     HL,(XYPOS)
0BC5 115000             LD     DE,80
0BC8 DD4E01             LD     C,(IX+1)
0BCB DD4602             LD     B,(IX+2)
0BCE 22280C             LD     (MAINL1+1),HL
0BD1 ED432B0C           LD     (MAINL2+1),BC      ・プログラムの初期化
0BD5 D9                 EXX
0BD6 2A0C01             LD     HL,(DDTOP)
0BD9 23                 INC    HL
0BDA 23                 INC    HL
0BDB 23                 INC    HL
0BDC 010000             LD     BC,0
0BDF D9                 EXX
0BE0 DD7E00             LD     A,(IX+0)
0BE3 32170C             LD     (BRGDT+1),A
0BE6 0E5C               LD     C,5CH
0BE8 21720C             LD     HL,SHIF1+1         ・B面の展開／表
0BEB 116C0C             LD     DE,GGDA1
0BEE CD0A0C             CALL   MAINL
0BF1 0E5D               LD     C,5DH
0BF3 21F40C             LD     HL,SHIF2+1         ・R面の展開／表示
0BF6 11ED0C             LD     DE,GGDA2
0BF9 CD0A0C             CALL   MAINL
0BFC 0E5E               LD     C,5EH
0BFE 21EE0D             LD     HL,SHIF3+1
0C01 11E70D             LD     DE,GGDA3           ・G面の展開／表示
0C04 CD0A0C             CALL   MAINL
0C07 D35F               OUT    (5FH),A
0C09 C9                 RET
```

```
0C0A ED79      MAINL:  OUT   (C),A
0C0C 79                LD    A,C
0C0D 32110D            LD    (CPON1+1),A      ・バンク切り換えとプログラムの書き換え
0C10 32170D            LD    (CPON2+1),A
0C13 328B0D            LD    (CPON3+1),A
0C16 3E00      BRGDT:  LD    A,0
0C18 0F                RRCA                   ・無データならNDISPへ
0C19 D2FD0E            JP    NC,NDISP
0C1C 0F                RRCA
0C1D 32170C            LD    (BRGDT+1),A
0C20 22320C            LD    (LPON1+1),HL
0C23 ED53510C          LD    (LPON2+1),DE     ・プログラム初期化
0C27 210000    MAINL1: LD    HL,0
0C2A 010000    MAINL2: LD    BC,0
0C2D 115000            LD    DE,80
0C30 C5        LOLOP:  PUSH  BC
0C31 3A720C    LPON1:  LD    A,(SHIF1+1)
0C34 E603              AND   00000011B
0C36 CA4F0C            JP    Z,LOLLP
0C39 D9                EXX
0C3A 3D                DEC   A
0C3B 280D              JR    Z,LPSF1
0C3D 3D                DEC   A
0C3E 2806              JR    Z,LPSF2
0C40 7A                LD    A,D
0C41 53                LD    D,E
0C42 5F                LD    E,A
0C43 C3500C            JP    LPON2
0C46 CB0B      LPSF2:  RRC   E
0C48 CB0A              RRC   D
0C4A CB0B      LPSF1:  RRC   E
0C4C CB0A              RRC   D
0C4E D9                EXX                    ・展開したデータを画面に表示する（縦列変更時は
0C4F D9        LOLLP:  EXX                      必要に応じDEをローテートする）
0C50 CD6C0C    LPON2:  CALL  GGDA1
0C53 D9                EXX
0C54 77                LD    (HL),A
0C55 19                ADD   HL,DE
0C56 10F7              DJNZ  LOLLP
0C58 23                INC   HL
0C59 C1                POP   BC
0C5A 0D                DEC   C
0C5B C8                RET   Z
0C5C 14                INC   D
0C5D 2006              JR    NZ,LPMHL
0C5F 1E50              LD    E,80
0C61 19                ADD   HL,DE
0C62 C3300C            JP    LOLOP
0C65 11B0FF    LPMHL:  LD    DE,-80
0C68 19                ADD   HL,DE
0C69 C3300C            JP    LOLOP

0C6C 78        GGDA1:  LD    A,B
0C6D B1                OR    C
0C6E 2816              JR    Z,NEWDT1
0C70 0B                DEC   BC
0C71 3E00      SHIF1:  LD    A,0
0C73 B7                OR    A
0C74 280E              JR    Z,DAFIX1
0C76 3D                DEC   A
0C77 2809              JR    Z,SFT11
0C79 3D                DEC   A                ・B面用データ展開プログラム
0C7A 2804              JR    Z,SFT12            BC＝圧縮データ連続数
0C7C 7A                LD    A,D                D＝前回のデータ
0C7D 53                LD    D,E
0C7E 5F                LD    E,A                E＝今回のデータ
0C7F C9                RET
0C80 CB0B      SFT12:  RRC   E
0C82 CB0B      SFT11:  RRC   E
0C84 7B        DAFIX1: LD    A,E
```

```
0C85 C9                 RET

0C86 7E        NEWDT1:  LD      A,(HL)
0C87 23                 INC     HL
0C88 5F                 LD      E,A
0C89 0F                 RRCA
0C8A 0F                 RRCA
0C8B 0F                 RRCA
0C8C 0F                 RRCA
0C8D BB                 CP      E
0C8E 7B                 LD      A,E
0C8F C0                 RET     NZ
0C90 3C                 INC     A
0C91 CAE80C             JP      Z,GSDTF1
0C94 3D                 DEC     A
0C95 CAD30C             JP      Z,GSDT01
0C98 FE55               CP      55H
0C9A CACE0C             JP      Z,GSD5A
0C9D FEAA               CP      0AAH
0C9F CACE0C             JP      Z,GSD5A
0CA2 C3B40C    DATAT1:  JP      DTYPA1
0CA5 FE33      DTYPB1:  CP      33H
0CA7 280B               JR      Z,DTYPA1
0CA9 FE66               CP      66H
0CAB 2807               JR      Z,DTYPA1          ・新しいデータに圧縮がかかっているかどうかを
0CAD FE99               CP      99H                 チェック
0CAF 2803               JR      Z,DTYPA1
0CB1 FECC               CP      0CCH
0CB3 C0                 RET     NZ
0CB4 3E02      DTYPA1:  LD      A,2
0CB6 4E        GSD5X1:  LD      C,(HL)
0CB7 23                 INC     HL
0CB8 CB79               BIT     7,C
0CBA 2803               JR      Z,CONT01+1
0CBC CBB9               RES     7,C
0CBE AF        CONT01:  XOR     A
0CBF 32720C             LD      (SHIF1+1),A
0CC2 3E7F               LD      A,7FH
0CC4 B9        STDCK0:  CP      C
0CC5 2004               JR      NZ,G0OK1
0CC7 46                 LD      B,(HL)
0CC8 23                 INC     HL
0CC9 4E                 LD      C,(HL)
0CCA 23                 INC     HL
0CCB 0B        G0OK1:   DEC     BC
0CCC 7B                 LD      A,E
0CCD C9                 RET

0CCE 3E01      GSD5A:   LD      A,1               ・圧縮サインが55H／AAHの場合の展開
0CD0 C3B60C             JP      GSD5X1

0CD3 7E        GSDT01:  LD      A,(HL)
0CD4 23                 INC     HL
0CD5 B7                 OR      A                 ・先頭圧縮サインが00Hの場合の展開
0CD6 2804               JR      Z,GS0001
0CD8 4F                 LD      C,A
0CD9 C3E00C             JP      GS00B1
0CDC 5E        GS0001:  LD      E,(HL)
0CDD 23                 INC     HL
0CDE 4E                 LD      C,(HL)
0CDF 23                 INC     HL
0CE0 AF        GS00B1:  XOR     A                 ・圧縮サインが00H，00Hの場合の展開
0CE1 32720C             LD      (SHIF1+1),A
0CE4 3D                 DEC     A
0CE5 C3C40C             JP      STDCK0

0CE8 4E        GSDTF1:  LD      C,(HL)
0CE9 23                 INC     HL                ・先頭圧縮サインがFFHの場合の展開
0CEA C3E00C             JP      GS00B1
```

```
0CED 78         GGDA2:  LD      A,B
0CEE B1                 OR      C
0CEF CA190D             JP      Z,NEWDT2
0CF2 0B                 DEC     BC
0CF3 3E00       SHIF2:  LD      A,0
0CF5 CB6F               BIT     5,A
0CF7 CA730C             JP      Z,SHIF1+2
```

・R面用データ展開プログラム（B面参照でなければSHIF1＋2へ）

```
0CFA D35C       BMENS2: OUT     (5CH),A
0CFC CB7F               BIT     7,A
0CFE C2130D             JP      NZ,RRB0
0D01 E603               AND     00000011B
0D03 2807               JR      Z,RRDB0
0D05 3D                 DEC     A
0D06 2802               JR      Z,RRDB1
0D08 CB0A               RRC     D
0D0A CB0A       RRDB1:  RRC     D
0D0C D9         RRDB0:  EXX
0D0D 7E                 LD      A,(HL)
0D0E D9                 EXX
0D0F A2                 AND     D
0D10 D35C       CPON1:  OUT     (5CH),A
0D12 C9                 RET
0D13 D9         RRB0:   EXX
0D14 7E                 LD      A,(HL)
0D15 D9                 EXX
0D16 D35C       CPON2:  OUT     (5CH),A
0D18 C9                 RET
```

・B面を参照してデータを得る場合

```
0D19 7E         NEWDT2: LD      A,(HL)
0D1A 23                 INC     HL
0D1B 5F                 LD      E,A
0D1C 0F                 RRCA
0D1D 0F                 RRCA
0D1E 0F                 RRCA
0D1F 0F                 RRCA
0D20 BB                 CP      E
0D21 7B                 LD      A,E
0D22 C0                 RET     NZ
0D23 3C                 INC     A
0D24 CA8D0D             JP      Z,GSDTF2
0D27 3D                 DEC     A
0D28 CA660D             JP      Z,GSDT02
0D2B FE55               CP      55H
0D2D CA9D0D             JP      Z,GSD5A2
0D30 FEAA               CP      0AAH
0D32 CA9D0D             JP      Z,GSD5A2
0D35 C3470D     DATAT2: JP      DTYPA2
0D38 FE33       DTYPB2: CP      33H
0D3A 280B               JR      Z,DTYPA2
0D3C FE66               CP      66H
0D3E 2807               JR      Z,DTYPA2
0D40 FE99               CP      99H
0D42 2803               JR      Z,DTYPA2
0D44 FECC               CP      0CCH
0D46 C0                 RET     NZ
0D47 4E         DTYPA2: LD      C,(HL)
0D48 3E02               LD      A,2
0D4A 23                 INC     HL
0D4B CB79               BIT     7,C
0D4D CA7B0D             JP      Z,CONT02
0D50 CBB9               RES     7,C
0D52 CB71               BIT     6,C
0D54 CAAC0D             JP      Z,BAND00
0D57 CBB1               RES     6,C
0D59 53                 LD      D,E
0D5A D9                 EXX
0D5B D35C               OUT     (5CH),A
0D5D 7E                 LD      A,(HL)
0D5E D9                 EXX
```

・新しいデータに圧縮がかかっているかどうかをチェック

```
0D5F A2                 AND    D
0D60 5F                 LD     E,A
0D61 3E22               LD     A,00100010B
0D63 C3D60D             JP     CONT12

0D66 7E        GSDT02:  LD     A,(HL)
0D67 23                 INC    HL
0D68 B7                 OR     A
0D69 CA910D             JP     Z,GS0002
0D6C 4F                 LD     C,A
0D6D E680               AND    10000000B
0D6F 280A               JR     Z,CONT02
0D71 CBB9               RES    7,C
0D73 D35C               OUT    (5CH),A
0D75 D9                 EXX
0D76 7E                 LD     A,(HL)
0D77 D9                 EXX
0D78 5F                 LD     E,A
0D79 3EA0               LD     A,10100000B        ・先頭圧縮サインが00Hの場合の展開
0D7B 32F40C    CONT02:  LD     (SHIF2+1),A
0D7E 3E7F               LD     A,7FH
0D80 B9        STDCK1:  CP     C
0D81 C2880D             JP     NZ,G0OK11
0D84 46                 LD     B,(HL)
0D85 23                 INC    HL
0D86 4E                 LD     C,(HL)
0D87 23                 INC    HL
0D88 0B        G0OK11:  DEC    BC
0D89 7B                 LD     A,E
0D8A D35C      CPON3:   OUT    (5CH),A
0D8C C9                 RET

0D8D 4E        GSDTF2:  LD     C,(HL)
0D8E 23                 INC    HL                 ・先頭圧縮サインがFFHの場合の展開
0D8F 1804               JR     GS00R2
0D91 5E        GS0002:  LD     E,(HL)
0D92 23                 INC    HL
0D93 4E                 LD     C,(HL)
0D94 23                 INC    HL                 ・圧縮サインが00H, 00Hの場合の展開
0D95 AF        GS00R2:  XOR    A
0D96 32F40C             LD     (SHIF2+1),A
0D99 3D                 DEC    A
0D9A C3800D             JP     STDCK1

0D9D 4E        GSD5A2:  LD     C,(HL)
0D9E 3E01               LD     A,1
0DA0 23                 INC    HL
0DA1 CB79               BIT    7,C
0DA3 CA7B0D             JP     Z,CONT02
0DA6 CBB9               RES    7,C
0DA8 CB71               BIT    6,C
0DAA 201E               JR     NZ,BMA52
0DAC CB69      BAND00:  BIT    5,C
0DAE 282E               JR     Z,RRRR0
0DB0 CB61               BIT    4,C
0DB2 282A               JR     Z,RRRR0
0DB4 CBA9               RES    5,C                ・圧縮サインが55H／AAHの場合の展開
0DB6 CBA1               RES    4,C
0DB8 53                 LD     D,E
0DB9 D9                 EXX
0DBA D35C               OUT    (5CH),A
0DBC 7E                 LD     A,(HL)
0DBD D9                 EXX
0DBE A2                 AND    D
0DBF 5F                 LD     E,A
0DC0 3E20               LD     A,00100000B
0DC2 32F40C             LD     (SHIF2+1),A
0DC5 3E0F               LD     A,0FH
0DC7 C3800D             JP     STDCK1
```

```
0DCA CBB1     BMA52:   RES   6,C
0DCC 53                LD    D,E
0DCD D9                EXX
0DCE D35C              OUT   (5CH),A
0DD0 7E                LD    A,(HL)
0DD1 D9                EXX                   ・1回目のデータが「B面 AND D」の場合
0DD2 A2                AND   D
0DD3 5F                LD    E,A
0DD4 3E21              LD    A,00100001B
0DD6 32F40C   CONT12:  LD    (SHIF2+1),A
0DD9 3E3F              LD    A,3FH
0DDB C3800D            JP    STDCK1
0DDE AF       RRRR0:   XOR   A
0DDF 32F40C            LD    (SHIF2+1),A     ・上位／下位が同じデータが連続している圧縮の展開
0DE2 3E2F              LD    A,2FH
0DE4 C3800D            JP    STDCK1

0DE7 78       GGDA3:   LD    A,B
0DE8 B1                OR    C
0DE9 CAFE0D            JP    Z,NEWDT3
0DEC 0B                DEC   BC
0DED 3E00     SHIF3:   LD    A,0
0DEF CB6F              BIT   5,A             ・G面用データ展開プログラム
0DF1 C2FA0C            JP    NZ,BMENS2
0DF4 CB67              BIT   4,A
0DF6 CA730C            JP    Z,SHIF1+2
0DF9 D35D              OUT   (5DH),A
0DFB C3FC0C            JP    BMENS2+2

0DFE 7E       NEWDT3:  LD    A,(HL)
0DFF 23                INC   HL
0E00 5F                LD    E,A
0E01 0F                RRCA
0E02 0F                RRCA
0E03 0F                RRCA
0E04 0F                RRCA
0E05 BB                CP    E
0E06 7B                LD    A,E
0E07 C0                RET   NZ
0E08 3C                INC   A
0E09 CA7E0E            JP    Z,GSDTF3
0E0C 3D                DEC   A
0E0D CA580E            JP    Z,GSDT03
0E10 FE55              CP    55H
0E12 CA920E            JP    Z,GSD5A3
0E15 FEAA              CP    0AAH            ・新しいデータに圧縮がかかっているかどうかを
0E17 CA920E            JP    Z,GSD5A3          チェック
0E1A C32C0E   DATAT3:  JP    DTYPA3
0E1D FE33     DTYPB3:  CP    33H
0E1F 280B              JR    Z,DTYPA3
0E21 FE66              CP    66H
0E23 2807              JR    Z,DTYPA3
0E25 FE99              CP    99H
0E27 2803              JR    Z,DTYPA3
0E29 FECC              CP    0CCH
0E2B C0                RET   NZ
0E2C 4E       DTYPA3:  LD    C,(HL)
0E2D 23                INC   HL
0E2E CB79              BIT   7,C
0E30 2005              JR    NZ,GXO81
0E32 3E02              LD    A,2
0E34 C39B0E            JP    GSDXX0
0E37 53       GXO81:   LD    D,E
0E38 CBB9              RES   7,C
0E3A CB71              BIT   6,C
0E3C 200C              JR    NZ,RMAXX
0E3E D9                EXX                   ・データが「B面 AND D」の連続の場合の展開
0E3F D35C              OUT   (5CH),A
0E41 7E                LD    A,(HL)            (Dは2ローテートする)
0E42 D9                EXX
```

```
0E43 A2                 AND     D
0E44 5F                 LD      E,A
0E45 3E22               LD      A,00100010B
0E47 C3E70E             JP      CONT13
```

```
0E4A CBB1      RMAXX:   RES     6,C
0E4C D9                 EXX
0E4D D35D               OUT     (5DH),A
0E4F 7E                 LD      A,(HL)
0E50 D9                 EXX
0E51 A2                 AND     D
0E52 5F                 LD      E,A
0E53 3E12               LD      A,00010010B
0E55 C3E70E             JP      CONT13
```

・データが「R面 AND D」の連続の場合の展開
（Dは２ローテートする）

```
0E58 7E        GSDT03:  LD      A,(HL)
0E59 23                 INC     HL
0E5A B7                 OR      A
0E5B 2815               JR      Z,GS0003
0E5D 4F                 LD      C,A
0E5E E680               AND     10000000B
0E60 280A               JR      Z,CONT03
0E62 CBB9               RES     7,C
0E64 D9                 EXX
0E65 D35C               OUT     (5CH),A
0E67 7E                 LD      A,(HL)
0E68 D9                 EXX
0E69 5F                 LD      E,A
0E6A 3EA0               LD      A,10100000B
0E6C 32EE0D    CONT03:  LD      (SHIF3+1),A
0E6F C37E0D             JP      STDCK1-2
```

・先頭圧縮サインが00Hの場合の展開

```
0E72 5E        GS0003:  LD      E,(HL)
0E73 23                 INC     HL
0E74 4E                 LD      C,(HL)
0E75 23                 INC     HL
0E76 AF                 XOR     A
0E77 32EE0D             LD      (SHIF3+1),A
0E7A 3D                 DEC     A
0E7B C3800D             JP      STDCK1
```

・データが交互連続の場合の展開

```
0E7E 4E        GSDTF3:  LD      C,(HL)
0E7F 23                 INC     HL
0E80 AF                 XOR     A
0E81 CB79               BIT     7,C
0E83 28E7               JR      Z,CONT03
0E85 CBB9               RES     7,C
0E87 D9                 EXX
0E88 D35D               OUT     (5DH),A
0E8A 7E                 LD      A,(HL)
0E8B D9                 EXX
0E8C 5F                 LD      E,A
0E8D 3E90               LD      A,10010000B
0E8F C36C0E             JP      CONT03
```

・先頭圧縮サインがFFHの場合の展開

```
0E92 4E        GSD5A3:  LD      C,(HL)
0E93 23                 INC     HL
0E94 CB79               BIT     7,C
0E96 C2D60E             JP      NZ,G5081
0E99 3E01               LD      A,1
0E9B CB71      GSDXX0:  BIT     6,C
0E9D CAE70E             JP      Z,CONT13
0EA0 CBB1               RES     6,C
0EA2 CB69               BIT     5,C
0EA4 CACD0E             JP      Z,GRRR0
0EA7 53                 LD      D,E
0EA8 CBA9               RES     5,C
0EAA CB61               BIT     4,C
0EAC 2011               JR      NZ,RMRR0
```

・連続数の数え方により分岐

```
0EAE D9                 EXX
0EAF D35C               OUT     (5CH),A
```

```
0EB1 7E                 LD      A,(HL)
0EB2 D9                 EXX
0EB3 A2                 AND     D                  ・データが「B面 AND D」の連続の場合の展開
0EB4 5F                 LD      E,A
0EB5 3E20               LD      A,00100000B
0EB7 32EE0D    CON00F:  LD      (SHIF3+1),A
0EBA 3E0F               LD      A,0FH
0EBC C3800D             JP      STDCK1
0EBF CBA1      RMRR0:   RES     4,C
0EC1 D9                 EXX
0EC2 D35D               OUT     (5DH),A
0EC4 7E                 LD      A,(HL)
0EC5 D9                 EXX                        ・データが「R面 AND D」の連続の場合の展開
0EC6 A2                 AND     D
0EC7 5F                 LD      E,A
0EC8 3E10               LD      A,00010000B
0ECA C3B70E             JP      CON00F
0ECD AF        GRRR0:   XOR     A
0ECE 32EE0D             LD      (SHIF3+1),A
0ED1 3E1F               LD      A,1FH              ・上位／下位が同じデータ（ローテートする）の場合
0ED3 C3800D             JP      STDCK1               の展開

0ED6 53        G5081:   LD      D,E
0ED7 CBB9               RES     7,C
0ED9 CB71               BIT     6,C
0EDB C2EF0E             JP      NZ,RMA55
0EDE D9                 EXX
0EDF D35C               OUT     (5CH),A
0EE1 7E                 LD      A,(HL)
0EE2 D9                 EXX                        ・データが「B面 AND D」の連続の場合の展開
0EE3 A2                 AND     D                   (Dは1ローテートする)
0EE4 5F                 LD      E,A
0EE5 3E21               LD      A,00100001B
0EE7 32EE0D    CONT13:  LD      (SHIF3+1),A
0EEA 3E3F               LD      A,3FH
0EEC C3800D             JP      STDCK1

0EEF CBB1      RMA55:   RES     6,C
0EF1 D9                 EXX
0EF2 D35D               OUT     (5DH),A
0EF4 7E                 LD      A,(HL)
0EF5 D9                 EXX                        ・データが「R面 AND D」の連続の場合の展開（Dは
0EF6 A2                 AND     D                   1ローテートする）
0EF7 5F                 LD      E,A
0EF8 3E11               LD      A,00010001B
0EFA C3E70E             JP      CONT13

0EFD 0F        NDISP:   RRCA
0EFE 32170C             LD      (BRGDT+1),A
0F01 2A0E01             LD      HL,(XYPOS)
0F04 DD4601             LD      B,(IX+1)
0F07 DD4E02             LD      C,(IX+2)
0F0A AF                 XOR     A
0F0B 115000             LD      DE,80
0F0E C5        NDIL0:   PUSH    BC
0F0F E5                 PUSH    HL
0F10 77        NDIL1:   LD      (HL),A             ・無データのプレーンを指定サイズでクリアする
0F11 23                 INC     HL
0F12 10FC               DJNZ    NDIL1
0F14 E1                 POP     HL
0F15 19                 ADD     HL,DE
0F16 C1                 POP     BC
0F17 0D                 DEC     C
0F18 20F4               JR      NZ,NDIL0
0F1A C9                 RET

                        ORG     0B700H

B700 C306B7             JP      PRESS              ・圧縮プログラム（テスト用）へ
B703 C322B7             JP      OPENS              ・展開／表示プログラム（テスト用）へ
```

```
B706 F3         PRESS:  DI
B707 3AC2E6             LD      A,(0E6C2H)
B70A F602               OR      2
B70C D331               OUT     (31H),A
B70E ED731EB7           LD      (PRSSP+1),SP
B712 3100B7             LD      SP,0B700H
B715 CD0001             CALL    100H
B718 3AC2E6             LD      A,(0E6C2H)
B71B D331               OUT     (31H),A
B71D 310000     PRSSP:  LD      SP,0
B720 FB                 EI
B721 C9                 RET
```

・圧縮プログラムをテストする

```
B722 F3         OPENS:  DI
B723 3AC2E6             LD      A,(0E6C2H)
B726 F602               OR      2
B728 D331               OUT     (31H),A
B72A ED733AB7           LD      (TOBSC+1),SP
B72E 3100B7             LD      SP,0B700H
B731 CD0301             CALL    103H
B734 3AC2E6             LD      A,(0E6C2H)
B737 D331               OUT     (31H),A
B739 310000     TOBSC:  LD      SP,0
B73C FB                 EI
B73D C9                 RET
```

・展開／表示プログラムをテストする



5-3 グラフィックス開発用支援ツール

「顔じゃないよ、心だよ……」と言いながらも、本心は美男美女に憧れているのが人間というものです。どんなにゲームのアイデアやプログラミング・テクニックが優れていても、それだけではゲームとしての魅力を満喫することはできません。ゲームの顔、それはやはり画面を構成するグラフィックスです。

これまでに紹介してきた種々のテクニックも、画面次第で価値は大きく変化します。いかにしてこの価値を高めるか、グラフィック・ツールの存在を無視してゲームを語ることはできないでしょう。これから紹介する各種ツールは、プロ用として私が開発したものを一般用にリメイクしたものです。めったに使われることのない特殊機能はカットしていますが、その分使い勝手を向上させてあります。市販のゲームと同等のグラフィックスを作成し、しかも本書で紹介したテクニックに合ったデータを得ることができます。大いに活用してください。

プログラムは、大型グラフィックス用の「PMAN88」、キャラクタ作成用の「PTER88」、スキャナコントロール用の「SCAN88」とに分かれていますが、これらは「GMENU」によって管理されていますから、エディタ上で自由に行き来ができます。なお、この他にもグラフィック・データ圧縮／展開用として、「5-2 グラフィック・データの圧縮と展開」で作成し「PRES88」のファイル名でセーブしたプログラムが必要です（添付ディスクにはセーブされています)。
プログラムの実行方法は次の通りです。

```
CLEAR,&HB5FF:BLOAD"GMENU",R
```

実行すると、メニュー画面（写真1）が現れますから、希望の番号を入力してください。なお、[B]を押してBASICへもどると、次回からは「ISET」とするだけで上記の内容を実行してくれます。

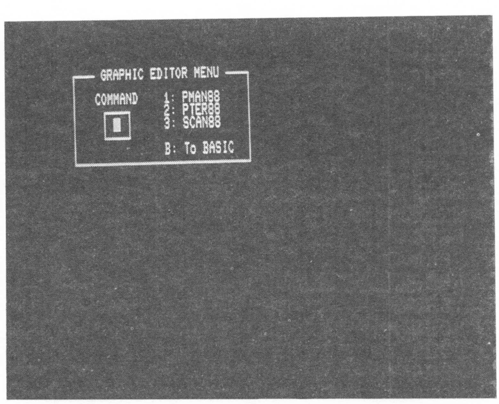


写真1　メニュー画面

【PMAN88】を選択すると……

画面は「PMAN88」のメニューに変わります。ここで HELP キーを押すと、このメニュー画面で実行できる内容が示されます（写真2）。それぞれの内容は次の通りです。コマンドからの回避は、特に指定のない限り STOP キーあるいは ESC キーでできます。

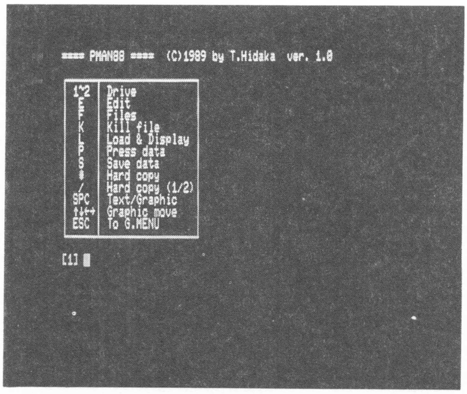


写真2　PMAN88のメニュー画面

1~2　：セーブ／ロードなどのファイルを扱う場合のドライブ番号です。

E　：グラフィックをエディットするモードへ移ります。

F　：指定されたドライブの全ファイル名を表示します。

K　：指定されたドライブにあるファイルを削除します。ファイル名を聞いてきますから、間違えないように入力してください。

L　：BSAVEされたファイルを表示しますから、番号で答えてください。データ形式が違うと「NOT DATA FILE !! 」となり表示はされません。ただし、圧縮されたデータかどうかを完全に判定することはできませんので、自分なりにファイル名に工夫を凝らして間違えないようにしてください。

P　：グラフィック画面のデータを圧縮します。画面には圧縮をかけるエリアを指定する枠が現れますので、テンキーにて調整してください。この時、左上は常に画面左上（G.VRAMのC000Hの位置）に固定されています。中央部を圧縮したい場合には、カーソルキー（ ↑ ↓ ← → ）にて画面を動かしてください。データ圧縮が完了すると、データアドレスと圧縮率が示されます。なお、横のサイズは8ドット単位となっています。

圧縮したデータが2000H～7FFFHに納まらない場合、残りのデータは無視されますので注意してください。すなわち、圧縮後にデータのエンドが7FFFHに近い場合（7FFCH以後）は、絵を分割して圧縮、セーブするようにしてください。

S　：圧縮されたデータをセーブしますから、ファイル名を入力してください。圧縮されたデータがない場合には、「SAVE DATA NOT

READY !! 」と表示されセーブできないようになっています。

＊　　：グラフィック画面をカラーハードコピーします。カラープリンタのスイッチを入れ、準備を整えておいてください。モノクロのプリンタの場合は、白以外はすべて黒になります。エリアの指定はデータ圧縮の時と同じですが、縦は4ライン単位となります。

／　　：グラフィック画面を1280ドットモードでカラーハードコピーします。つまり、全体が1/2に縮小されてコピーされるわけですが、このモードでは縦ラインが1ラインづつスキップされたようになります。また、エリア指定の縦は8ライン単位となります。

SPC　：スペースキーを押すたびに、テキスト画面／グラフィック画面が交互に切り替わります。

↑↓←→：グラフィック画面を上下左右に動かします。左右の場合、**TAB** キーを押しながら移動させるとドット単位に動かすことができます。

ESC　：メニュー画面（写真1）にもどります。

では、**E** キーを押して画面エディットモード（写真3）へ入ってみましょう。画面右に、グラフィックスを拡大したものが表示され、その下にはエディット中の状態や使用できる主要なコマンドを示す欄があります。コマンドによっては確認や質問をしてくるものがありますが、それらは枠で囲まれたメッセージ欄に現れます。カーソルの移動はテンキーによって行い、現在のドット位置は「・＝ (0, 0)」の部分に示されます。**SHIFT** キーを押すと高速に移動できます。

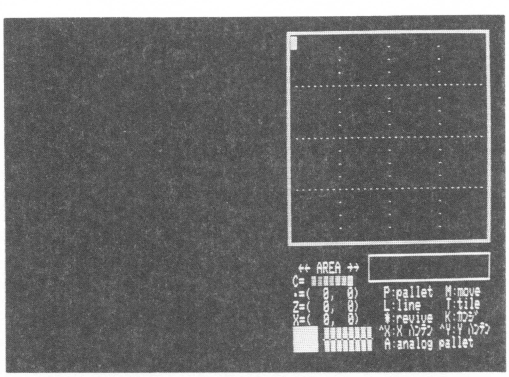


写真3　PMAN88のエディット画面

エディタ部におけるコマンド

C　：色の指定。カラー番号 0〜8で指定します。カラー番号=8というのは、下部に示されるタイリングカラーのことです。なお、ドットのセットは[5]キーまたはスペースキー、リセットは[0]キーまたは[GRPH]キーです。

Z　：ドット位置の記憶-①

X　：ドット位置の記憶-②

P　：パレットの変更（デジタル)。

A　：パレットの変更（アナログ)。カラー番号を入力すると、現在のBRG各面についての濃度が示されますから、それぞれについて 0〜7の数値を入力してください。最後まで入力せずに[⏎]キーを押した場合は変更されません。

M　：Z-Xで囲まれたグラフィックスを、現在カーソルのある位置に移動（コピー）します。

L　：Z-Xを結ぶラインまたはボックスを描きます。

T　：タイリングカラーによるペイントを行います。通常のペイント、ボックスフィルの他にスーパータイリングという特殊ペイントがあります。これは、Z-Xで囲まれたエリアにある指定の一色（または黒以外の全色）をタイリングカラーで塗るというものです。タイリングカラーの指定は、実際に色を確認しながら作成することができます。なお、ペイント中にペイント洩れを起こした場合など、[STOP]キーでペイントを中断することができます。

＊　：拡大表示されている部分に限り、初期状態（そのボックス内に入った時の状態）にもどすことができます。

K　：漢字表示。漢字コード表を見てコード番号で入力してください。

^X　：Z-Xで囲まれた部分を左右反転します。

^Y　：Z-Xで囲まれた部分を上下反転します。

←→　：エディットエリアの変更。右側の隠された部分をエディットしたい場合に使用しますが、ペイントや移動などすべてのコマンドは画面に表示されているエリアに対してのみ行われます。そのため、Z-Xによる指定も連動して動きますから注意してください。

カナ　：[カナ]キーをロックすると、拡大されている部分がボックスで示され、移動もボックス単位となります。

H.CLR　：カーソルを左上に移動します。[カナ]キーロックの場合は拡大ボックスを左上に移動させます。

／，．　：[H.CLR]キーと同様に、カーソルあるいは拡大ボックスを、それぞれ右上、左下、右下に移動させます。

ESC　：［PMAN88］のメニューにもどります。

【PTER88】を選択すると……

画面は「PTER88」のメニューに変わります。ここで HELP キーを押すと、このメニュー画面で実行できる内容が示されます（写真 4）。

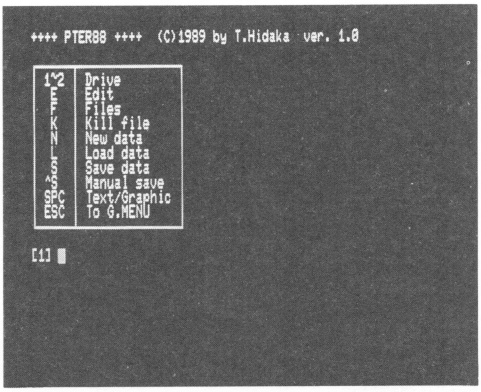


写真 4　PTER88のメニュー画面

それぞれの内容は次の通りです。

1～2　：セーブ／ロードなどのファイルを扱う場合のドライブ番号です。

E　：キャラクタ・パターンをエディットするモードへ移ります。

F　：指定されたドライブの全ファイル名を表示します。

K　：指定されたドライブにあるファイルを削除します。

N　：データをクリアします。

L　：BSAVE されたファイルを表示しますから、番号で答えてください。

S　：データをセーブしますから、ファイル名を入力してください。

^S　：アドレスを指定してデータをセーブします。特定のパターンデータだけセーブしたい場合に便利です。

SPC　：スペースキーを押すたびに、テキスト画面／グラフィック画面が交互に切り替わります。

ESC　：メニュー画面（写真 1）にもどります。

では、E キーを押して画面エディットモード（写真 5）へ入ってみましょう。画面左に、パターンを拡大したものが表示されます。［PMAN88］とほとんど同じですが、キャラクタ・パターン用ということでいくつかの違いがあります。

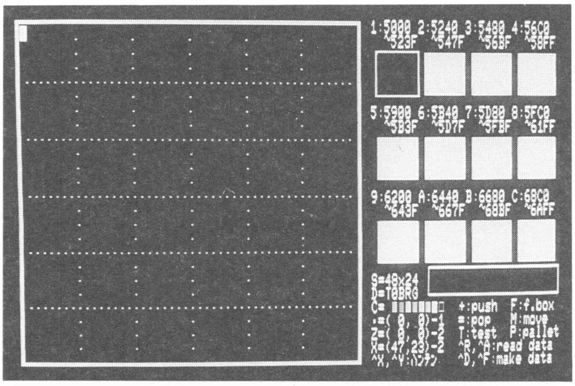


写真 5　PTER88のエディット画面

エディタ部におけるコマンド

- **S** ：キャラクタ・パターンのサイズを示しています。作成したいパターンのサイズに合わせてください。
- **D** ：データのタイプを示しています。通常のBRGデータと、くり抜きデータを含んだものがあります。ALUを利用した重ね合わせの場合は、透明部分のビットがゼロとなりますから「T0BRG」を選択します。ALUのない機種では、透明部分のビットが1となる「T1BRG」が重ね合わせ用のデータです。
- **C** ：色の指定。カラー番号0〜8で指定します。カラー番号＝8というのは、重ね合わせに使うくり抜き部分を意味します。パレットを変更するように色を変えることができます。ドットのセット／リセットは[PMAN88] と同じです。
- **Z** ：ドット位置の記憶-①
- **X** ：ドット位置の記憶-②
- **^X** ：Z-Xで囲まれた部分を左右反転します。
- **^Y** ：Z-Xで囲まれた部分を上下反転します。
- **+** ：エディット中のパターンを記憶します。
- **=** ：記憶してあるパターンを再生します。
- **F** ：Z-Xで囲まれた部分をカラー番号で示された色で塗りつぶします。
- **M** ：Z-Xで囲まれたグラフィックスを、現在カーソルのある位置に移動（コピー）します。
- **T** ：パターンのアニメーション・テストを行います。最初にアニメーションの順序を入力し、次に好みのウェイトを入力してください。なお、アニメーション中はスペースキーで一時停止することができます。
- **P** ：パレットの変更（デジタル)。
- **^R** ：エディット中のパターン・データをリードし画面上に表示します。
- **^A** ：すべてのパターン・データをリードし画面上に表示します。
- **^D** ：エディット中のパターンをデータ化します。
- **^F** ：すべてのパターンをデータ化します。
- **カナ** ：カナキーをロックすると、エディット・パターンを上下左右に移動させることができます。
- **H.CLR** ：カーソルを左上に移動します。
- **/, .** ：それぞれカーソルを右上、左下、右下に移動します。
- **ESC** ：[PTER88] のメニューにもどります。

【SCAN88】を選択すると……

画面は「SCAN88」のコントロール画面（写真6）に変わります。スキャナは種類によって機能が違いますから、必ず機種を合わせてください。主要な部分は機種設定によって選択肢が制限されるようになっています。ただし、スキャナは細かな部分における仕様の違いが結構多いため、プログラム側ですべてについて正確に判断しているわけではありません。そのため、サイズやDPIなどについてはマニュアルでは不可能な設定もできることがあるかもしれませんが、それらは無視されますので注意してください。なお、各機能の設定はDPIの一部を除いてすべてロータリー式になっています。

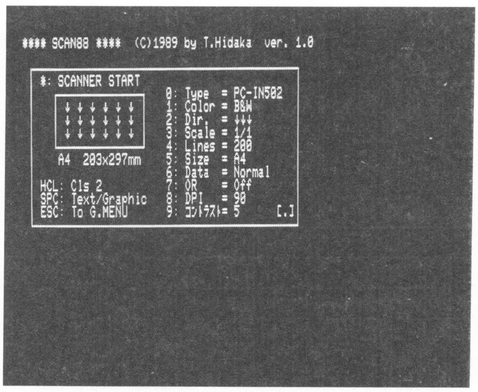


写真6　SCAN88のコントロール画面

また、実際にスキャナを使用する場合には、ボーレート等の通信条件を本体とスキャナ側で一致させる必要があります。本ソフトウェアは、下記の条件で動作を確認しておりますので最初に本体／スキャナの設定を行ってください。☆印は、スキャナが本体側に要求している設定です。ボーレートについては、19200ボーでは異常が発生する危険性がありますので（特にカラーの場合）、必ず9600ボー以下に設定してください。

【RS-232C通信条件】

☆ボーレート	：9600ボー
通信方式	：全二重
Xパラメータ	：無効
☆ストップビット長	：1ビット
☆データビット長	：8ビット
Sパラメータ	：無効
DELコード	：有効
☆パリティチェック	：無し
パリティ	：偶数

[SCAN88] におけるコマンド

0 ：スキャナの機種設定。こちらで登録してあるのは次の各機種ですが、ここに示されたもの以外でもマニュアルに「機能は……に準ずる」とたいてい表示されているはずですから、それに合わせてお使いください。
PC-IN502, PC-IN503, PC-IN503H, PC-IN505, PC-IN506
PC-PR406S, PC-PR406HS（以上、日本電気）
HS7R, HS7R2, HS10R, HS10R2（以上、オムロン）

1 ：モノクロ、カラー（面順次／線順次）の選択。

2 ：画面表示方向の選択（縦／横）。

3 ：画面表示の際の縮小率（1/1, 1/2）。

4 ：画面のモード選択（640×200, 640×400）。ただし、640×400は参考としてあるだけで、これをエディットしたりセーブする機能は付いていません。

5 ：読み取り部のサイズを指定します。

6 ：読み取ったデータをそのまま表示するか、白黒を反転して表示するかを選択します。通常はノーマルです。

7 ：スキャナ側で読み取ったデータを縮小する場合、間引かれたデータを隣接するデータとORを取って残すかどうかの選択です。ただし、機種によってはこの機能を無視しますので、マニュアルで確認してください。

8 ：読み取り線密度の指定。90～240DPIの範囲で変化しますが、PC-IN506のカラー線順次に限り5～99DPIのマニュアル設定（←→による）もできるようにしてあります。

9 ：モノクロモードでは常にコントラストですが、カラーモードではブライト、色合い、彩度と変化します。それぞれ[・]キーにより0～9の範囲で指定することができます。

***** ：スキャナ・スタート。スキャナによっては、本体側でもスタートスイッチを押さなければならないものもあります。なお、[STOP]キーにより途中でスキャナを停止させることができます。

H.CLR ：画面をクリアします（CLS 2)。

SPC ：スペースキーを押すたびに、テキスト画面／グラフィック画面が交互に切り替わります。

ESC ：メニュー画面（写真1）にもどります。

以上が、本書で用意したグラフィック・ツールの全機能です。一度にすべての機能を覚えるのは大変かもしれませんが、ゲーム制作においてはどれ1つとして欠かすことができないものばかりです。ぜひ、うまく使いこなして市販ソフトを越えるゲームにチャレンジしてください。

※エディタ・アセンブラ『HIT-88』(ver. 1.2) について

筆者が開発したエディタ・アセンブラ『HIT-88』は、初心者からプロの方まで幅広くご使用いただけるよう、便利さと機能性を兼ね備えています。お求めになりたい方は、現金書留（または郵便小為替＋簡易書留）にて下記宛申し込んでください。価格は送料込みで2000円（消費税不要）です。

〒353 埼玉県志木市館1-6-14-103
新日本プログラミング／日高　徹

申し込みの有効期限は本書の発行日より１年間です。それ以降につきましては、バージョン／価格等に変更がないかどうか、最初に往復ハガキでご確認ください。ただし、『ポプコム』等のパソコン誌上に記事として明示してある場合は、その必要はありません。

なお、『HIT-88』(ver. 1.2) はPC-8801シリーズ全機種に対応していますが、メディアは５”２Ｄのみで２HDや８”のディスケットには対応しておりません。

付録 マシン語インストラクション一覧表

ここに掲載するマシン語インストラクション一覧表は、日本電気株式会社の「μCOM-82インストラクション活用表」より、転載いたしました。

[注1] ニモニック中の略号の意味

略　号	意　　味
d, n	8ビット・イミーディエート・データ
e	相対アドレシングの変位置
e－2	eの実効変位置

[注2] フラグの記号の意味

記　号		意　　味
•		影響受けない
0		リセット
1		セット
×		不定
↕		演算結果に従った影響を受ける（①②③あり＝下に説明）
P		1＝偶数パリティ，0＝奇数パリティ
V		1＝オーバーフローあり,0＝オーバーフローなし
IFF		P／Vフラグ← (IFF)
↕	①	BC－1＝0ならばP／V＝0, その他P／V＝1
	②	A＝（HL）ならばZ＝1, その他Z＝0
	③	B－1＝0ならばZ＝1, その他Z＝0

[注3] OPコードの注記の意味

Ⓐ		Ⓑ		Ⓒ		Ⓓ		Ⓔ		Ⓕ		Ⓖ				Ⓗ	
	ss																
Reg	dd	Reg	qq	Reg	pp	Reg	rr	Reg	r,r'	Bit	b	cc		Condition	Flag	P	t
BC	00	BC	00	BC	00	BC	00	B	000	0	000	000	NZ	Non Zero	Z	00H	000
DE	01	DE	01	DE	01	DE	01	C	001	1	001	001	Z	Zero	Z	08H	001
HL	10	HL	10	IX	10	IY	10	D	010	2	010	010	NC	Non Carry	C	10H	010
SP	11	AF	11	SP	11	SP	11	E	011	3	011	011	C	Carry	C	18H	011
								H	100	4	100	100	PO	Parity Odd	P/V	20H	100
								L	101	5	101	101	PE	Parity Even	P/V	28H	101
								A	111	6	110	110	P	Sign Positive	S	30H	110
										7	111	111	M	Sign Negative	S	38H	111

[1] インストラクション・セット

命令群	ニモニック	オペレーション	フラグ						バイト	ステート	OPコード			
			S	Z	H	P/V	N	C			76	543	210	
8ビット・ロード命令	LD r, r'	r←r'	•	•	•	•	•	•	1	4	01	r	r'	Ⓔ
	LD r, n	r←n	•	•	•	•	•	•	2	7	00	r	110	Ⓔ
											←	n	→	
	LD r, (HL)	r← (HL)	•	•	•	•	•	•	1	7	01	r	110	Ⓔ
	LD r, (IX＋d)	r← (IX＋d)	•	•	•	•	•	•	3	19	11	011	101	
											01	r	110	Ⓔ
											←	d	→	
	LD r, (IY＋d)	r← (IY＋d)	•	•	•	•	•	•	3	19	11	111	101	
											01	r	110	Ⓔ
											←	d	→	
	LD (HL), r	(HL) ←r	•	•	•	•	•	•	1	7	01	110	r	Ⓔ
	LD (IX＋d), r	(IX＋d) ←r	•	•	•	•	•	•	3	19	11	011	101	
											01	110	r	Ⓔ
											←	d	→	
	LD (IY＋d), r	(IY＋d) ←r	•	•	•	•	•	•	3	19	11	111	101	
											01	110	r	Ⓔ
											←	d	→	

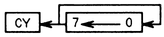
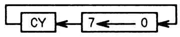
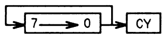
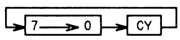
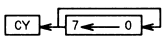
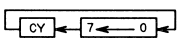
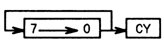
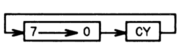
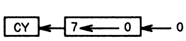
命令群	ニモニック	オペレーション	フラグ S	Z	H	P/V	N	C	バイト	ステート	OPコード 76	543	210	
8ビット・ロード命令	LD (HL), n	(HL) ←n	•	•	•	•	•	•	2	10	00	110	110	
											←	n	→	
	LD (IX+d), n	(IX+d) ←n	•	•	•	•	•	•	4	19	11	011	101	
											00	110	110	
											←	d	→	
											←	n	→	
	LD (IY+d), n	(IY+d) ←n	•	•	•	•	•	•	4	19	11	111	101	
											00	110	110	
											←	d	→	
											←	n	→	
	LD A, (BC)	A← (BC)	•	•	•	•	•	•	1	7	00	001	010	
	LD A, (DE)	A← (DE)	•	•	•	•	•	•	1	7	00	011	010	
	LD A, (nn)	A← (nn)	•	•	•	•	•	•	3	13	00	111	010	
											←	n	→	
											←	n	→	
	LD (BC), A	(BC) ←A	•	•	•	•	•	•	1	7	00	000	010	
	LD (DE), A	(DE) ←A	•	•	•	•	•	•	1	7	00	010	010	
	LD (nn), A	(nn) ←A	•	•	•	•	•	•	3	13	00	110	010	
											←	n	→	
											←	n	→	
	LD A, I	A←I	↕	↕	0	IFF	0	•	2	9	11	101	101	
											01	010	111	
	LD A, R	A←R	↕	↕	0	IFF	0	•	2	9	11	101	101	
											01	011	111	
	LD I, A	I←A	•	•	•	•	•	•	2	9	11	101	101	
											01	000	111	
	LD R, A	R←A	•	•	•	•	•	•	2	9	11	101	101	
											01	001	111	
16ビット・ロード命令	LD dd, nn	dd←nn	•	•	•	•	•	•	3	10	00	dd0	011	Ⓐ
											←	n	→	
											←	n	→	
	LD IX, nn	IX←nn	•	•	•	•	•	•	4	14	11	011	101	
											00	100	001	
											←	n	→	
											←	n	→	
	LD IY, nn	IY←nn	•	•	•	•	•	•	4	14	11	111	101	
											00	100	001	
											←	n	→	
											←	n	→	
	LD HL, (nn)	H← (nn+1)	•	•	•	•	•	•	3	16	00	101	010	
		L← (nn)									←	n	→	
											←	n	→	
	LD dd, (nn)	dd_H← (nn+1)	•	•	•	•	•	•	4	20	11	101	101	
		dd_L← (nn)									01	dd1	011	Ⓐ
											←	n	→	
											←	n	→	
	LD IX, (nn)	IX_H← (nn+1)	•	•	•	•	•	•	4	20	11	011	101	
		IX_L← (nn)									00	101	010	
											←	n	→	
											←	n	→	
	LD IY, (nn)	IY_H← (nn+1)	•	•	•	•	•	•	4	20	11	111	101	
		IY_L← (nn)									00	101	010	
											←	n	→	
											←	n	→	
	LD (nn), HL	(nn+1) ←H	•	•	•	•	•	•	3	16	00	100	010	
		(nn) ←L									←	n	→	
											←	n	→	
	LD (nn), dd	(nn+1) ←dd_H	•	•	•	•	•	•	4	20	11	101	101	
		(nn) ←dd_L									01	dd0	011	Ⓐ
											←	n	→	
											←	n	→	
	LD (nn), IX	(nn+1) ←IX_H	•	•	•	•	•	•	4	20	11	011	101	
		(nn) ←IX_L									00	100	010	
											←	n	→	
											←	n	→	

命令群	ニモニック	オペレーション	S	Z	H	P/V	N	C	バイト	ステート	76	543	210	
16ビット・ロード命令	LD (nn), IY	(nn + 1) ← IY_H	•	•	•	•	•	•	4	20	11	111	101	
		(nn) ← IY_L									00	100	010	
											←	n	→	
											←	n	→	
	LD SP, HL	SP ← HL	•	•	•	•	•	•	1	6	11	111	001	
	LD SP, IX	SP ← IX	•	•	•	•	•	•	2	10	11	011	101	
											11	111	001	
	LD SP, IY	SP ← IY	•	•	•	•	•	•	2	10	11	111	101	
											11	111	001	
	PUSH qq	(SP − 2) ← qq_L	•	•	•	•	•	•	1	11	11	qq0	101	Ⓑ
		(SP − 1) ← qq_H												
	PUSH IX	(SP − 2) ← IX_L	•	•	•	•	•	•	2	15	11	011	101	
		(SP − 1) ← IX_H									11	100	101	
	PUSH IY	(SP − 2) ← IY_L	•	•	•	•	•	•	2	15	11	111	101	
		(SP − 1) ← IY_H									11	100	101	
	POP qq	qq_H ← (SP + 1)	•	•	•	•	•	•	1	10	11	qq0	001	Ⓑ
		qq_L ← (SP)												
	POP IX	IX_H ← (SP + 1)	•	•	•	•	•	•	2	14	11	011	101	
		IX_L ← (SP)									11	100	001	
	POP IY	IY_H ← (SP + 1)	•	•	•	•	•	•	2	14	11	111	101	
		IY_L ← (SP)									11	100	001	
エクスチェンジ命令	EX DE, HL	DE ↔ HL	•	•	•	•	•	•	1	4	11	101	011	
	EX AF, AF'	AF ↔ AF'	•	•	•	•	•	•	1	4	00	001	000	
	EXX	(BC ↔ BC'	•	•	•	•	•	•	1	4	11	011	001	
		DE ↔ DE'												
		HL ↔ HL')												
	EX (SP), HL	H ↔ (SP + 1)	•	•	•	•	•	•	1	19	11	100	011	
		L ↔ (SP)												
	EXX (SP), IX	IX_H ↔ (SP + 1)	•	•	•	•	•	•	2	23	11	011	101	
		IX_L ↔ (SP)									11	100	011	
	EX (SP), IY	IY_H ↔ (SP + 1)	•	•	•	•	•	•	2	23	11	111	101	
		IY_L ↔ (SP)									11	100	011	
ブロック転送命令	LDI	(DE) ← (HL)	•	•	0	① ↕	0	•	2	16	11	101	101	
		DE ← DE + 1									10	100	000	
		HL ← HL + 1												
		BC ← BC − 1												
	LDIR	(DE) ← (HL)	•	•	0	0	0	•	2	21 if BC ≠ 0	11	101	101	
		DE ← DE + 1								16 if BC = 0	10	110	000	
		HL ← HL + 1												
		BC ← BC − 1 until BC = 0												
	LDD	(DE) ← (HL)	•	•	0	① ↕	0	•	2	16	11	101	101	
		DE ← DE − 1									10	101	000	
		HL ← HL − 1												
		BC ← BC − 1												
	LDDR	(DE) ← (HL)	•	•	0	0	0	•	2	21 if BC = 0	11	101	101	
		DE ← DE − 1								16 if BC = 0	10	111	000	
		HL ← HL − 1												
		BC ← BC − 1 until BC = 0												
ブロック・サーチ命令	CPI	A − (HL)	↕	② ↕	↕	① ↕	1	•	2	16	11	101	101	
		HL ← HL + 1									10	100	001	
		BC ← BC − 1												
	CPIR	A − (HL)	↕	② ↕	↕	① ↕	1	•	2	21 if BC ≠ 0	11	101	101	
		HL ← HL + 1								and A ≠ (HL)	10	110	001	
		BC ← BC − 1 until								16 if BC = 0				
		A = (HL) or BC = 0								or A = (HL)				
	CPD	A − (HL)	↕	② ↕	↕	① ↕	1	•	2	16	11	101	101	
		HL ← HL − 1									10	101	001	
		BC ← BC − 1												
	CPDR	A − (HL)	↕	② ↕	↕	① ↕	1	•	2	21 if BC ≠ 0	11	101	101	
		HL ← HL − 1								and A ≠ (HL)	10	111	001	
		BC ← BC − 1 until								16 if BC = 0				
		A = (HL) or BC = 0								or A = (HL)				

(Column groups: フラグ = S, Z, H, P/V, N, C; OPコード = 76, 543, 210)

命令群	ニモニック	オペレーション	フラグ S	Z	H	P/V	N	C	バイト	ステート	OPコード 76	543	210	
8ビット算術論理演算命令	ADD A, r	A←A+r	↕	↕	↕	V	0	↕	1	4	10	000	r	Ⓔ
	ADD A, n	A←A+n	↕	↕	↕	V	0	↕	2	7	11	000	110	
											←	n	→	
	ADD A,(HL)	A←A+ (HL)	↕	↕	↕	V	0	↕	1	7	10	000	110	
	ADD A,(IX+d)	A←A+ (IX+d)	↕	↕	↕	V	0	↕	3	19	11	011	101	
											10	000	110	
											←	d	→	
	ADD A,(IY+d)	A←A+ (IY+d)	↕	↕	↕	V	0	↕	3	19	11	111	101	
											10	000	110	
											←	d	→	
	ADC A, r	A←A+r+CY	↕	↕	↕	V	0	↕	1	4	10	001	r	Ⓔ
	ADC A, n	A←A+n+CY	↕	↕	↕	V	0	↕	2	7	11	001	110	
											←	n	→	
	ADC A,(HL)	A←A+ (HL) +CY	↕	↕	↕	V	0	↕	1	7	10	001	110	
	ADC A,(IX+d)	A←A+ (IX+d) +CY	↕	↕	↕	V	0	↕	3	19	11	011	101	
											10	001	110	
											←	d	→	
	ADC A,(IY+d)	A←A+ (IY+d) +CY	↕	↕	↕	V	0	↕	3	19	11	111	101	
											10	001	110	
											←	d	→	
	SUB r	A←A−r	↕	↕	↕	V	1	↕	1	4	10	010	r	Ⓔ
	SUB n	A←A−n	↕	↕	↕	V	1	↕	2	7	11	010	110	
											←	n	→	
	SUB (HL)	A←A− (HL)	↕	↕	↕	V	1	↕	1	7	10	010	110	
	SUB (IX+d)	A←A− (IX+d)	↕	↕	↕	V	1	↕	3	19	11	011	101	
											10	010	110	
											←	d	→	
	SUB (IY+d)	A←A− (IY+d)	↕	↕	↕	V	1	↕	3	19	11	111	101	
											10	010	110	
											←	d	→	
	SBC A, r	A←A−r−CY	↕	↕	↕	V	1	↕	1	4	10	011	r	Ⓔ
	SBC A, n	A←A−n−CY	↕	↕	↕	V	1	↕	2	7	11	011	110	
											←	n	→	
	SBC A,(HL)	A←A− (HL) −CY	↕	↕	↕	V	1	↕	1	7	10	011	110	
	SBC A,(IX+d)	A←A− (IX+d) −CY	↕	↕	↕	V	1	↕	3	19	11	011	101	
											10	011	110	
											←	d	→	
	SBC A,(IY+d)	A←A− (IY+d) −CY	↕	↕	↕	V	1	↕	3	19	11	111	101	
											10	011	110	
											←	d	→	
	AND r	A←A∧r	↕	↕	1	P	0	0	1	4	10	100	r	Ⓔ
	AND n	A←A∧n	↕	↕	1	P	0	0	2	7	11	100	110	
											←	n	→	
	AND (HL)	A←A∧ (HL)	↕	↕	1	P	0	0	1	7	10	100	110	
	AND (IX+d)	A←A∧ (IX+d)	↕	↕	1	P	0	0	3	19	11	011	101	
											10	100	110	
											←	d	→	
	AND (IY+d)	A←A∧ (IY+d)	↕	↕	1	P	0	0	3	19	11	111	101	
											10	100	110	
											←	d	→	
	OR r	A←A∨r	↕	↕	0	P	0	0	1	4	10	110	r	Ⓔ
	OR n	A←A∨n	↕	↕	0	P	0	0	2	7	11	110	110	
											←	n	→	
	OR (HL)	A←A∨ (HL)	↕	↕	0	P	0	0	1	7	10	110	110	
	OR (IX+d)	A←A∨ (IX+d)	↕	↕	0	P	0	0	3	19	11	011	101	
											10	110	110	
											←	d	→	
	OR (IY+d)	A←A∨ (IY+d)	↕	↕	0	P	0	0	3	19	11	111	101	
											10	110	110	
											←	d	→	
	XOR r	A←A⊻r	↕	↕	0	P	0	0	1	4	10	101	r	Ⓔ
	XOR n	A←A⊻n	↕	↕	0	P	0	0	2	7	11	101	110	
											←	n	→	
	XOR (HL)	A←A⊻ (HL)	↕	↕	0	P	0	0	1	7	10	101	110	
	XOR (IX+d)	A←A⊻ (IX+d)	↕	↕	0	P	0	0	3	19	11	011	101	
											10	101	110	
											←	d	→	

命令群	ニモニック	オペレーション	フラグ S	Z	H	P/V	N	C	バイト	ステート	OPコード 76	543	210	
8ビット算術論理演算命令	XOR (IY+d)	A←A ⊻ (IY+d)	↕	↕	0	P	0	0	3	19	11	111	101	
											10	101	110	
											←	d	→	
	CP r	A−r	↕	↕	↕	V	1	↕	1	4	10	111	r	Ⓔ
	CP n	A−n	↕	↕	↕	V	1	↕	2	7	11	111	110	
											←	n	→	
	CP (HL)	A− (HL)	↕	↕	↕	V	1	↕	1	7	10	111	110	
	CP (IX+d)	A− (IX+d)	↕	↕	↕	V	1	↕	3	19	11	011	101	
											10	111	110	
											←	d	→	
	CP (IY+d)	A− (IY+d)	↕	↕	↕	V	1	↕	3	19	11	111	101	
											10	111	110	
											←	d	→	
	INC r	r←r+1	↕	↕	↕	V	0	•	1	4	00	r	100	Ⓔ
	INC (HL)	(HL) ← (HL) +1	↕	↕	↕	V	0	•	1	11	00	110	100	
	INC (IX+d)	(IX+d) ← (IX+d) +1	↕	↕	↕	V	0	•	3	23	11	011	101	
											00	110	100	
											←	d	→	
	INC (IY+d)	(IY+d) ← (IY+d) +1	↕	↕	↕	V	0	•	3	23	11	111	101	
											00	110	100	
											←	d	→	
	DEC r	r←r−1	↕	↕	↕	V	1	•	1	4	00	r	101	Ⓔ
	DEC (HL)	(HL) ← (HL) −1	↕	↕	↕	V	1	•	1	11	00	110	101	
	DEC (IX+d)	(IX+d) ← (IX+d) −1	↕	↕	↕	V	1	•	3	23	11	011	101	
											00	110	101	
											←	d	→	
	DEC (IY+d)	(IY+d) ← (IY+d) −1	↕	↕	↕	V	1	•	3	23	11	111	101	
											00	110	101	
											←	d	→	
16ビット算術演算命令	ADD HL, ss	HL←HL+ss	•	•	×	•	0	↕	1	11	00	ss1	001	Ⓐ
	ADC HL, ss	HL←HL+ss+CY	↕	↕	×	V	0	↕	2	15	11	101	101	
											01	ss1	010	Ⓐ
	SBC HL, ss	HL←HL−ss−CY	↕	↕	×	V	1	↕	2	15	11	101	101	
											01	ss0	010	Ⓐ
	ADD IX, pp	IX←IX+pp	•	•	×	•	0	↕	2	15	11	011	101	
											00	pp1	001	Ⓒ
	ADD IY, rr	IY←IY+rr	•	•	×	•	0	↕	2	15	11	111	101	
											00	rr1	001	Ⓓ
	INC ss	ss←ss+1	•	•	•	•	•	•	1	6	00	ss0	011	Ⓐ
	INC IX	IX←IX+1	•	•	•	•	•	•	2	10	11	011	101	
											00	100	011	
	INC IY	IY←IY+1	•	•	•	•	•	•	2	10	11	111	101	
											00	100	011	
	DEC ss	ss←ss−1	•	•	•	•	•	•	1	6	00	ss1	011	Ⓐ
	DEC IX	IX←IX−1	•	•	•	•	•	•	2	10	11	011	101	
											00	101	011	
	DEC IY	IY←IY−1	•	•	•	•	•	•	2	10	11	111	101	
											00	101	011	
アキュムレータ操作命令	DAA	Decimal Adjust Acc	↕	↕	↕	P	•	↕	1	4	00	100	111	
	CPL	$A \leftarrow \overline{A}$	•	•	1	•	1	•	1	4	00	101	111	
	NEG	$A \leftarrow \overline{A}+1$	↕	↕	↕	V	1	↕	2	8	11	101	101	
											01	000	100	
	CCF	$CY \leftarrow \overline{CY}$	•	•	×	•	0	↕	1	4	00	111	111	
	SCF	CY←1	•	•	0	•	0	1	1	4	00	110	111	
CPUコントロール命令	NOP	No operation	•	•	•	•	•	•	1	4	00	000	000	
	HALT	CPU halted	•	•	•	•	•	•	1	4	01	110	110	
	DI	IFF←0	•	•	•	•	•	•	1	4	11	110	011	
	EI	IFF←1	•	•	•	•	•	•	1	4	11	111	011	
	IM 0	Set interrupt mode 0	•	•	•	•	•	•	2	8	11	101	101	
											01	000	110	
	IM 1	Set interrupt mode 1	•	•	•	•	•	•	2	8	11	101	101	
											01	010	110	
	IM 2	Set interrupt mode 2	•	•	•	•	•	•	2	8	11	101	101	
											01	011	110	

命令群	ニモニック	オペレーション	フラグ S	Z	H	P/V	N	C	バイト	ステート	OPコード 76	543	210	
ローテート・シフト命令	RLCA		•	•	0	•	0	↕	1	4	00	000	111	
	RLA		•	•	0	•	0	↕	1	4	00	010	111	
	RRCA		•	•	0	•	0	↕	1	4	00	001	111	
	RRA		•	•	0	•	0	↕	1	4	00	011	111	
	RLC r	 r, (HL), (IX + d),(IY + d)	↕	↕	0	P	0	↕	2	8	11 00	001 000	011 r	Ⓔ
	RLC (HL)		↕	↕	0	P	0	↕	2	15	11 00	001 000	011 110	
	RLC (IX + d)		↕	↕	0	P	0	↕	4	23	11 11 ← 00	011 001 d 000	101 011 → 110	
	RLC (IY + d)		↕	↕	0	P	0	↕	4	23	11 11 ← 00	111 001 d 000	101 011 → 110	
	RL r	 r, (HL), (IX + d),(IY + d)	↕	↕	0	P	0	↕	2	8	11 00	001 010	011 r	Ⓔ
	RL (HL)		↕	↕	0	P	0	↕	2	15	11 00	001 010	011 110	
	RL (IX + d)		↕	↕	0	P	0	↕	4	23	11 11 ← 00	011 001 d 010	101 011 → 110	
	RL (IY + d)		↕	↕	0	P	0	↕	4	23	11 11 ← 00	111 001 d 010	101 011 → 110	
	RRC r	 r, (HL),(IX + d),(IY + d)	↕	↕	0	P	0	↕	2	8	11 00	001 001	011 r	Ⓔ
	RRC (HL)		↕	↕	0	P	0	↕	2	15	11 00	001 001	011 110	
	RRC (IX + d)		↕	↕	0	P	0	↕	4	23	11 11 ← 00	011 001 d 001	101 011 → 110	
	RRC (IY + d)		↕	↕	0	P	0	↕	4	23	11 11 ← 00	111 001 d 001	101 011 → 110	
	RR r	 r, (HL),(IX + d),(IY + d)	↕	↕	0	P	0	↕	2	8	11 00	001 011	011 r	Ⓔ
	RR (HL)		↕	↕	0	P	0	↕	2	15	11 00	001 011	011 110	
	RR (IX + d)		↕	↕	0	P	0	↕	4	23	11 11 ← 00	011 001 d 011	101 011 → 110	
	RR (IY + d)		↕	↕	0	P	0	↕	4	23	11 11 ← 00	111 001 d 011	101 011 → 110	
	SLA r	 r, (HL),(IX + d),(IY + d)	↕	↕	0	P	0	↕	2	8	11 00	001 100	011 r	Ⓔ
	SLA (HL)		↕	↕	0	P	0	↕	2	15	11 00	001 100	011 110	
	SLA (IX + d)		↕	↕	0	P	0	↕	4	23	11 11 ← 00	011 001 d 100	101 011 → 110	

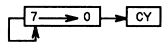
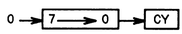
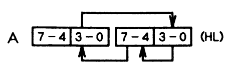
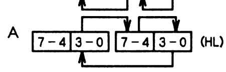
命令群	ニモニック	オペレーション	S	Z	H	P/V	N	C	バイト	ステート	76	543	210	
ローテート・シフト命令	SLA (IY + d)		↕	↕	0	P	0	↕	4	23	11	111	101	
											11	001	011	
											←	d	→	
											00	100	110	
	SRA r		↕	↕	0	P	0	↕	2	8	11	001	011	
											00	101	r	Ⓔ
	SRA (HL)		↕	↕	0	P	0	↕	2	15	11	001	011	
											00	101	110	
	SRA (IX + d)		↕	↕	0	P	0	↕	4	23	11	011	101	
		r, (HL),(IX + d),(IY + d)									11	001	011	
											←	d	→	
											00	101	110	
	SRA (IY + d)		↕	↕	0	P	0	↕	4	23	11	111	101	
											11	001	011	
											←	d	→	
											00	101	110	
	SRL r		↕	↕	0	P	0	↕	2	8	11	001	011	
											00	111	r	Ⓔ
	SRL (HL)		↕	↕	0	P	0	↕	2	15	11	001	011	
											00	111	110	
	SRL (IX + d)		↕	↕	0	P	0	↕	4	23	11	011	101	
		r, (HL),(IX + d),(IY + d)									11	001	011	
											←	d	→	
											00	111	110	
	SRL (IY + d)		↕	↕	0	P	0	↕	4	23	11	111	101	
											11	001	011	
											←	d	→	
											00	111	110	
	RLD		↕	↕	0	P	0	•	2	18	11	101	101	
											01	101	111	
	RRD		↕	↕	0	P	0	•	2	18	11	101	101	
											01	100	111	
ビット操作命令	BIT b, r	Z←$\overline{r}_b$	×	↕	1	×	0	•	2	8	11	001	011	
											01	b	r	ⒺⒻ
	BIT b, (HL)	Z← $(\overline{HL})_b$	×	↕	1	×	0	•	2	12	11	001	011	
											01	b	110	Ⓕ
	BIT b, (IX + d)	Z← $(\overline{IX+d})_b$	×	↕	1	×	0	•	4	20	11	011	101	
											11	001	011	
											←	d	→	
											01	b	110	Ⓕ
	BIT b, (IY + d)	Z← $(\overline{IY+d})_b$	×	↕	1	×	0	•	4	20	11	111	101	
											11	001	011	
											←	d	→	
											01	b	110	Ⓕ
	SET b, r	r_b←1	•	•	•	•	•	•	2	8	11	001	011	
											11	b	r	ⒺⒻ
	SET b, (HL)	$(HL)_b$←1	•	•	•	•	•	•	2	15	11	001	011	
											11	b	110	Ⓕ
	SET b, (IX + d)	$(IX + d)_b$←1	•	•	•	•	•	•	4	23	11	011	101	
											11	001	011	
											←	d	→	
											11	b	110	Ⓕ
	SET b, (IY + d)	$(IY + d)_b$←1	•	•	•	•	•	•	4	23	11	111	101	
											11	001	011	
											←	d	→	
											11	b	110	Ⓕ
	RES b, r	r_b←0	•	•	•	•	•	•	2	8	11	001	011	
											10	b	r	ⒺⒻ
	RES b, (HL)	$(HL)_b$←0	•	•	•	•	•	•	2	15	11	001	011	
											10	b	110	Ⓕ
	RES b, (IX + d)	$(IX + d)_b$←0	•	•	•	•	•	•	4	23	11	011	101	
											11	001	011	
											←	d	→	
											10	b	110	Ⓕ
	RES b, (IY + d)	$(IY + d)_b$←0	•	•	•	•	•	•	4	23	11	111	101	
											11	001	011	
											←	d	→	
											10	b	110	Ⓕ

Table header groups: フラグ (S, Z, H, P/V, N, C); OP コード (76, 543, 210).

命令群	ニモニック	オペレーション	フラグ S	Z	H	P/V	N	C	バイト	ステート	OPコード 76	543	210	
ジャンプ・コール・リターン命令	JP nn	PC←nn	•	•	•	•	•	•	3	10	11 ← ←	000 n n	011 → →	
	JP cc, nn	If cc is true PC←nn otherwise continue	•	•	•	•	•	•	3	10	11 ← ←	cc n n	010 → →	Ⓖ
	JR e	PC←PC+e	•	•	•	•	•	•	2	12	00 ←	011 e-2	000 →	
	JR C, e	If C=0 continue If C=1 PC←PC+e	•	•	•	•	•	•	2	7 if C=0 12 if C=1	00 ←	111 e-2	000 →	
	JR NC, e	if C=1 continue if C=0 PC←PC+e	•	•	•	•	•	•	2	7 if C=1 12 if C=0	00 ←	110 e-2	000 →	
	JR Z, e	if Z=0 continue if Z=1 PC←PC+e	•	•	•	•	•	•	2	7 if Z=0 12 if Z=1	00 ←	101 e-2	000 →	
	JR NZ, e	if Z=1 continue if Z=0 PC←PC+e	•	•	•	•	•	•	2	7 if Z=1 12 if Z=0	00 ←	100 e-2	000 →	
	JP (HL)	PC←HL	•	•	•	•	•	•	1	4	11	101	001	
	JP (IX)	PC←IX	•	•	•	•	•	•	2	8	11 11	011 101	101 001	
	JP (IY)	PC←IY	•	•	•	•	•	•	2	8	11 11	111 101	101 001	
	DJNZ e	B←B-1 if B=0 continue if B≠0 PC←PC+e	•	•	•	•	•	•	2	8 if B=0 13 if B≠0	00 ←	010 e-2	000 →	
	CALL nn	(SP−1)←PC_H (SP−2)←PC_L PC←nn	•	•	•	•	•	•	3	17	11 ← ←	001 n n	101 → →	
	CALL cc, nn	If cc is false continue otherwise same as CALL nn	•	•	•	•	•	•	3	10 if cc is false 17 if cc is true	11 ← ←	cc n n	100 → →	Ⓖ
	RET	PC_L←(SP) PC_H←(SP+1)	•	•	•	•	•	•	1	10	11	001	001	
	RET cc	If cc is false continue othewise same as RET	•	•	•	•	•	•	1	5 if cc is false 11 if cc is true	11	cc	000	Ⓖ
	RETI	Return from interrupt	•	•	•	•	•	•	2	14	11 01	101 001	101 101	
	RETN	Return from non maskable interrupt	•	•	•	•	•	•	2	14	11 01	101 000	101 101	
	RST p	(SP−1)←PC_H (SP−2)←PC_L PC_H←0 PC_L←p	•	•	•	•	•	•	1	11	11	t	111	Ⓗ
入出力命令	IN A, n	A←(n) A_{0-7}←n A_{8-15}←A	•	•	•	•	•	•	2	11	11 ←	011 n	011 →	
	IN r, (C)	r←(C) if r=110 only the flags will be affected A_{0-7}←C A_{8-15}←B	↕	↕	↕	P	0	•	2	12	11 01	101 r	101 000	Ⓔ
	INI	(HL)←(C) B←B−1 HL←HL+1 A_{0-7}←C A_{8-15}←B	×	③ ↕	×	×	1	•	2	16	11 10	101 100	101 010	
	INIR	(HL)←(C) B←B−1 HL←HL+1 until B=0 A_{0-7}←C A_{8-15}←B	X	1	X	X	1	•	2	21 if B≠0 16 if B=0	11 10	101 110	101 010	
	IND	(HL)←(C) B←B−1 HL←HL−1 A_{0-7}←C A_{8-15}←B	X	③ ↕	X	X	1	•	2	16	11 10	101 101	101 010	

命令群	ニモニック	オペレーション	S	Z	H	P/V	N	C	バイト	ステート	76	543	210	
入出力命令	INDR	(HL) ← (C) B←B − 1 HL←HL − 1 unitl B = 0 A_{0-7}←C A_{8-15}←B	X	1	X	X	1	•	2	21 if B≠0 16 if B = 0	11 10	101 111	101 010	
	OUT n, A	(n) ←A A_{0-7}←M A_{8-15}←B	•	•	•	•	•	•	2	11	11 ←	101 n	011 →	
	OUT (C), r	(C) ←r A_{0-7}←C A_{8-15}←B	•	•	•	•	•	•	2	12	11 01	101 r	101 001	Ⓔ
	OUTI	(C) ← (HL) B←B − 1 HL←HL + 1 A_{0-7}←C A_{8-15}←B	X	③ ↕	X	X	1	•	2	16	11 10	101 100	101 011	
	OTIR	(C) ← (HL) B←B − 1 HL←HL + 1 until B = 0 A_{0-7}←C A_{8-15}←B	X	1	X	X	1	•	2	21 if B≠0 16 if B = 0	11 10	101 110	101 011	
	OUTD	(C) ← (HL) B←B − 1 HL←HL − 1 A_{0-7}←C A_{8-15}←B	X	③ ↕	X	X	1	•	2	16	11 10	101 101	101 011	
	OTDR	(C) ← (HL) B←B − 1 HL←HL − 1 until B = 0 A_{0-7}←C A_{8-15}←B	X	1	X	X	1	•	2	21 if B≠0 16 if B = 0	11 10	101 111	101 011	

[2] マシン語←→ニモニック対応表

00	NOP		40	LD	B, B	80	ADD	A, B	C0	RET	NZ
01	LD	BC, nn	41	LD	B, C	81	ADD	A, C	C1	POP	BC
02	LD	(BC), A	42	LD	B, D	82	ADD	A, D	C2	JP	NZ, nn
03	INC	BC	43	LD	B, E	83	ADD	A, E	C3	JP	nn
04	INC	B	44	LD	B, H	84	ADD	A, H	C4	CALL	NZ, nn
05	DEC	B	45	LD	B, L	85	ADD	A, L	C5	PUSH	BC
06	LD	B, n	46	LD	B, (HL)	86	ADD	A, (HL)	C6	ADD	A, n
07	RLCA		47	LD	B, A	87	ADD	A, A	C7	RST	00H
08	EX	AF, AF'	48	LD	C, B	88	ADC	A, B	C8	RET	Z
09	ADD	HL, BC	49	LD	C, C	89	ADC	A, C	C9	RET	
0A	LD	A, (BC)	4A	LD	C, D	8A	ADC	A, D	CA	JP	Z, nn
0B	DEC	BC	4B	LD	C, E	8B	ADC	A, E	CB	□	
0C	INC	C	4C	LD	C, H	8C	ADC	A, H	CC	CALL	Z, nn
0D	DEC	C	4D	LD	C, L	8D	ADC	A, L	CD	CALL	nn
0E	LD	C, n	4E	LD	C, (HL)	8E	ADC	A, (HL)	CE	ADC	A, n
0F	RRCA		4F	LD	C, A	8F	ADC	A, A	CF	RST	08H
10	DJNZ	e	50	LD	D, B	90	SUB	B	D0	RET	NC
11	LD	DE, nn	51	LD	D, C	91	SUB	C	D1	POP	DE
12	LD	(DE), A	52	LD	D, D	92	SUB	D	D2	JP	NC, nn
13	INC	DE	53	LD	D, E	93	SUB	E	D3	OUT	n, A
14	INC	D	54	LD	D, H	94	SUB	H	D4	CALL	NC, nn
15	DEC	D	55	LD	D, L	95	SUB	L	D5	PUSH	DE
16	LD	D, n	56	LD	D, (HL)	96	SUB	(HL)	D6	SUB	n
17	RLA		57	LD	D, A	97	SUB	A	D7	RST	10H
18	JR	e	58	LD	E, B	98	SBC	A, B	D8	RET	C
19	ADD	HL, DE	59	LD	E, C	99	SBC	A, C	D9	EXX	
1A	LD	A, (DE)	5A	LD	E, D	9A	SBC	A, D	DA	JP	C, nn
1B	DEC	DE	5B	LD	E, E	9B	SBC	A, E	DB	IN	A, n
1C	INC	E	5C	LD	E, H	9C	SBC	A, H	DC	CALL	C, nn
1D	DEC	E	5D	LD	E, L	9D	SBC	A, L	DD	□	
1E	LD	E, n	5E	LD	E, (HL)	9E	SBC	A, (HL)	DE	SBC	A, n
1F	RRA		5F	LD	E, A	9F	SBC	A, A	DF	RST	18H
20	JR	NZ, e	60	LD	H, B	A0	AND	B	E0	RET	PO
21	LD	HL, nn	61	LD	H, C	A1	AND	C	E1	POP	HL
22	LD	(nn), HL	62	LD	H, D	A2	AND	D	E2	JP	PO, nn
23	INC	HL	63	LD	H, E	A3	AND	E	E3	EX	(SP), HL
24	INC	H	64	LD	H, H	A4	AND	H	E4	CALL	PO, nn
25	DEC	H	65	LD	H, L	A5	AND	L	E5	PUSH	HL
26	LD	H, n	66	LD	H, (HL)	A6	AND	(HL)	E6	AND	n
27	DAA		67	LD	H, A	A7	AND	A	E7	RST	20H
28	JR	Z, e	68	LD	L, B	A8	XOR	B	E8	RET	PE
29	ADD	HL, HL	69	LD	L, C	A9	XOR	C	E9	JP	(HL)
2A	LD	HL, (nn)	6A	LD	L, D	AA	XOR	D	EA	JP	PE, nn
2B	DEC	HL	6B	LD	L, E	AB	XOR	E	EB	EX	DE, HL
2C	INC	L	6C	LD	L, H	AC	XOR	H	EC	CALL	PE, nn
2D	DEC	L	6D	LD	L, L	AD	XOR	L	ED	□	
2E	LD	L, n	6E	LD	L, (HL)	AE	XOR	(HL)	EE	XOR	n
2F	CPL		6F	LD	L, A	AF	XOR	A	EF	RST	28H
30	JR	NC, e	70	LD	(HL), B	B0	OR	B	F0	RET	P
31	LD	SP, nn	71	LD	(HL), C	B1	OR	C	F1	POP	AF
32	LD	(nn), A	72	LD	(HL), D	B2	OR	D	F2	JP	P, nn
33	INC	SP	73	LD	(HL), E	B3	OR	E	F3	DI	
34	INC	(HL)	74	LD	(HL), H	B4	OR	H	F4	CALL	P, nn
35	DEC	(HL)	75	LD	(HL), L	B5	OR	L	F5	PUSH	AF
36	LD	(HL), n	76	HALT		B6	OR	(HL)	F6	OR	n
37	SCF		77	LD	(HL), A	B7	OR	A	F7	RST	30H
38	JR	C, e	78	LD	A, B	B8	CP	B	F8	RET	M
39	ADD	HL, SP	79	LD	A, C	B9	CP	C	F9	LD	SP, HL
3A	LD	A, (nn)	7A	LD	A, D	BA	CP	D	FA	JP	M, nn
3B	DEC	SP	7B	LD	A, E	BB	CP	E	FB	EI	
3C	INC	A	7C	LD	A, H	BC	CP	H	FC	CALL	M, nn
3D	DEC	A	7D	LD	A, L	BD	CP	L	FD	□	
3E	LD	A, n	7E	LD	A, (HL)	BE	CP	(HL)	FE	CP	n
3F	CCF		7F	LD	A, A	BF	CP	A	FF	RST	38H

CB ××

00	RLC	B	40	BIT	0, B	80	RES	0, B	C0	SET	0, B
01	RLC	C	41	BIT	0, C	81	RES	0, C	C1	SET	0, C
02	RLC	D	42	BIT	0, D	82	RES	0, D	C2	SET	0, D
03	RLC	E	43	BIT	0, E	83	RES	0, E	C3	SET	0, E
04	RLC	H	44	BIT	0, H	84	RES	0, H	C4	SET	0, H
05	RLC	L	45	BIT	0, L	85	RES	0, L	C5	SET	0, L
06	RLC	(HL)	46	BIT	0, (HL)	86	RES	0, (HL)	C6	SET	0, (HL)
07	RLC	A	47	BIT	0, A	87	RES	0, A	C7	SET	0, A
08	RRC	B	48	BIT	1, B	88	RES	1, B	C8	SET	1, B
09	RRC	C	49	BIT	1, C	89	RES	1, C	C9	SET	1, C
0A	RRC	D	4A	BIT	1, D	8A	RES	1, D	CA	SET	1, D
0B	RRC	E	4B	BIT	1, E	8B	RES	1, E	CB	SET	1, E
0C	RRC	H	4C	BIT	1, H	8C	RES	1, H	CC	SET	1, H
0D	RRC	L	4D	BIT	1, L	8D	RES	1, L	CD	SET	1, L
0E	RRC	(HL)	4E	BIT	1, (HL)	8E	RES	1, (HL)	CE	SET	1, (HL)
0F	RRC	A	4F	BIT	1, A	8F	RES	1, A	CF	SET	1, A
10	RL	B	50	BIT	2, B	90	RES	2, B	D0	SET	2, B
11	RL	C	51	BIT	2, C	91	RES	2, C	D1	SET	2, C
12	RL	D	52	BIT	2, D	92	RES	2, D	D2	SET	2, D
13	RL	E	53	BIT	2, E	93	RES	2, E	D3	SET	2, E
14	RL	H	54	BIT	2, H	94	RES	2, H	D4	SET	2, H
15	RL	L	55	BIT	2, L	95	RES	2, L	D5	SET	2, L
16	RL	(HL)	56	BIT	2, (HL)	96	RES	2, (HL)	D6	SET	2, (HL)
17	RL	A	57	BIT	2, A	97	RES	2, A	D7	SET	2, A
18	RR	B	58	BIT	3, B	98	RES	3, B	D8	SET	3, B
19	RR	C	59	BIT	3, C	99	RES	3, C	D9	SET	3, C
1A	RR	D	5A	BIT	3, D	9A	RES	3, D	DA	SET	3, D
1B	RR	E	5B	BIT	3, E	9B	RES	3, E	DB	SET	3, E
1C	RR	H	5C	BIT	3, H	9C	RES	3, H	DC	SET	3, H
1D	RR	L	5D	BIT	3, L	9D	RES	3, L	DD	SET	3, L
1E	RR	(HL)	5E	BIT	3, (HL)	9E	RES	3, (HL)	DE	SET	3, (HL)
1F	RR	A	5F	BIT	3, A	9F	RES	3, A	DF	SET	3, A
20	SLA	B	60	BIT	4, B	A0	RES	4, B	E0	SET	4, B
21	SLA	C	61	BIT	4, C	A1	RES	4, C	E1	SET	4, C
22	SLA	D	62	BIT	4, D	A2	RES	4, D	E2	SET	4, D
23	SLA	E	63	BIT	4, E	A3	RES	4, E	E3	SET	4, E
24	SLA	H	64	BIT	4, H	A4	RES	4, H	E4	SET	4, H
25	SLA	L	65	BIT	4, L	A5	RES	4, L	E5	SET	4, L
26	SLA	(HL)	66	BIT	4, (HL)	A6	RES	4, (HL)	E6	SET	4, (HL)
27	SLA	A	67	BIT	4, A	A7	RES	4, A	E7	SET	4, A
28	SRA	B	68	BIT	5, B	A8	RES	5, B	E8	SET	5, B
29	SRA	C	69	BIT	5, C	A9	RES	5, C	E9	SET	5, C
2A	SRA	D	6A	BIT	5, D	AA	RES	5, D	EA	SET	5, D
2B	SRA	E	6B	BIT	5, E	AB	RES	5, E	EB	SET	5, E
2C	SRA	H	6C	BIT	5, H	AC	RES	5, H	EC	SET	5, H
2D	SRA	L	6D	BIT	5, L	AD	RES	5, L	ED	SET	5, L
2E	SRA	(HL)	6E	BIT	5, (HL)	AE	RES	5, (HL)	EE	SET	5, (HL)
2F	SRA	A	6F	BIT	5, A	AF	RES	5, A	EF	SET	5, A
30			70	BIT	6, B	B0	RES	6, B	F0	SET	6, B
31			71	BIT	6, C	B1	RES	6, C	F1	SET	6, C
32			72	BIT	6, D	B2	RES	6, D	F2	SET	6, D
33			73	BIT	6, E	B3	RES	6, E	F3	SET	6, E
34			74	BIT	6, H	B4	RES	6, H	F4	SET	6, H
35			75	BIT	6, L	B5	RES	6, L	F5	SET	6, L
36			76	BIT	6, (HL)	B6	RES	6, (HL)	F6	SET	6, (HL)
37			77	BIT	6, A	B7	RES	6, A	F7	SET	6, A
38	SRL	B	78	BIT	7, B	B8	RES	7, B	F8	SET	7, B
39	SRL	C	79	BIT	7, C	B9	RES	7, C	F9	SET	7, C
3A	SRL	D	7A	BIT	7, D	BA	RES	7, D	FA	SET	7, D
3B	SRL	E	7B	BIT	7, E	BB	RES	7, E	FB	SET	7, E
3C	SRL	H	7C	BIT	7, H	BC	RES	7, H	FC	SET	7, H
3D	SRL	L	7D	BIT	7, L	BD	RES	7, L	FD	SET	7, L
3E	SRL	(HL)	7E	BIT	7, (HL)	BE	RES	7, (HL)	FE	SET	7, (HL)
3F	SRL	A	7F	BIT	7, A	BF	RES	7, A	FF	SET	7, A

DD ××

09			ADD	IX, BC
19			ADD	IX, DE
21			LD	IX, nn
22			LD	(nn), IX
23			INC	IX
29			ADD	IX, IX
2A			LD	IX, (nn)
2B			DEC	IX
34			INC	(IX + d)
35			DEC	(IX + d)
36			LD	(IX + d), n
39			ADD	IX, SP
46			LD	B, (IX + d)
4E			LD	C, (IX + d)
56			LD	D, (IX + d)
5E			LD	E, (IX + d)
66			LD	H, (IX + d)
6E			LD	L, (IX + d)
70			LD	(IX + d), B
71			LD	(IX + d), C
72			LD	(IX + d), D
73			LD	(IX + d), E
74			LD	(IX + d), H
75			LD	(IX + d), L
77			LD	(IX + d), A
7E			LD	A, (IX + d)
86			ADD	A, (IX + d)
8E			ADC	A, (IX + d)
96			SUB	(IX + d)
9E			SBC	A, (IX + d)
A6			AND	(IX + d)
AE			XOR	(IX + d)
B6			OR	(IX + d)
BE			CP	(IX + d)
CB	d	06	RLC	(IX + d)
CB	d	0E	RRC	(IX + d)
CB	d	16	RL	(IX + d)
CB	d	1E	RR	(IX + d)
CB	d	26	SLA	(IX + d)
CB	d	2E	SRA	(IX + d)
CB	d	3E	SRL	(IX + d)
CB	d	46	BIT	0, (IX + d)
CB	d	4E	BIT	1, (IX + d)
CB	d	56	BIT	2, (IX + d)
CB	d	5E	BIT	3, (IX + d)
CB	d	66	BIT	4, (IX + d)
CB	d	6E	BIT	5, (IX + d)
CB	d	76	BIT	6, (IX + d)
CB	d	7E	BIT	7, (IX + d)
CB	d	86	RES	0, (IX + d)
CB	d	8E	RES	1, (IX + d)
CB	d	96	RES	2, (IX + d)
CB	d	9E	RES	3, (IX + d)
CB	d	A6	RES	4, (IX + d)
CB	d	AE	RES	5, (IX + d)
CB	d	B6	RES	6, (IX + d)
CB	d	BE	RES	7, (IX + d)
CB	d	C6	SET	0, (IX + d)
CB	d	CE	SET	1, (IX + d)
CB	d	D6	SET	2, (IX + d)
CB	d	DE	SET	3, (IX + d)
CB	d	E6	SET	4, (IX + d)
CB	d	EE	SET	5, (IX + d)
CB	d	F6	SET	6, (IX + d)
CB	d	FE	SET	7, (IX + d)
E1			POP	IX
E3			EX	(SP), IX
E5			PUSH	IX
E9			JP	(IX)
F9			LD	SP, IX

ED ××

40	IN	B, (C)
41	OUT	(C), B
42	SBC	HL, BC
43	LD	(nn), BC
44	NEG	
45	RETN	
46	IM	0
47	LD	I, A
48	IN	C, (C)
49	OUT	(C), C
4A	ADC	HL, BC
4B	LD	BC, (nn)
4D	RETI	
4F	LD	R, A
50	IN	D, (C)
51	OUT	(C), D
52	SBC	HL, DE
53	LD	(nn), DE
56	IM	1
57	LD	A, I
58	IN	E, (C)
59	OUT	(C), E
5A	ADC	HL, DE
5B	LD	DE, (nn)
5E	IM	2
5F	LD	A, R
60	IN	h, (C)
61	OUT	(C), H
62	SBC	HL, HL
67	RRD	
68	IN	L, (C)
69	OUT	(C), L
6A	ADC	HL, HL
6F	RLD	
72	SBC	HL, SP
73	LD	(nn), SP
78	IN	A, (C)
79	OUT	(C), A
7A	ADC	HL, SP
7B	LD	SP, (nn)
A0	LDI	
A1	CPI	
A2	INI	
A3	OUTI	
A8	LDD	
A9	CPD	
AA	IND	
AB	OUTD	
B0	LDIR	
B1	CPIR	
B2	INIR	
B3	OTIR	
B8	LDDR	
B9	CPDR	
BA	INDR	
BB	OTDR	

FD ××

09			ADD	IY, BC
19			ADD	IY, DE
21			LD	IY, nn
22			LD	(nn), IY
23			INC	IY
29			ADD	IY, IY
2A			LD	IY, (nn)
2B			DEC	IY
34			INC	(IY + d)
35			DEC	(IY + d)
36			LD	(IY + d), n
39			ADD	IY, SP
46			LD	B, (IY + d)
4E			LD	C, (IY + d)
56			LD	D, (IY + d)
5E			LD	E, (IY + d)
66			LD	H, (IY + d)
6E			LD	L, (IY + d)
70			LD	(IY + d), B
71			LD	(IY + d), C
72			LD	(IY + d), D
73			LD	(IY + d), E
74			LD	(IY + d), H
75			LD	(IY + d), L
77			LD	(IY + d), A
7E			LD	A, (IY + d)
86			ADD	A, (IY + d)
8E			ADC	A, (IY + d)
96			SUB	(IY + d)
9E			SBC	A, (IY + d)
A6			AND	(IY + d)
AE			XOR	(IY + d)
B6			OR	(IY + d)
BE			CP	(IY + d)
CB	d	06	RLC	(IY + d)
CB	d	0E	RRC	(IY + d)
CB	d	16	RL	(IY + d)
CB	d	1E	RR	(IY + d)
CB	d	26	SLA	(IY + d)
CB	d	2E	SRA	(IY + d)
CB	d	3E	SRL	(IY + d)
CB	d	46	BIT	0, (IY + d)
CB	d	4E	BIT	1, (IY + d)
CB	d	56	BIT	2, (IY + d)
CB	d	5E	BIT	3, (IY + d)
CB	d	66	BIT	4, (IY + d)
CB	d	6E	BIT	5, (IY + d)
CB	d	76	BIT	6, (IY + d)
CB	d	7E	BIT	7, (IY + d)
CB	d	86	RES	0, (IY + d)
CB	d	8E	RES	1, (IY + d)
CB	d	96	RES	2, (IY + d)
CB	d	9E	RES	3, (IY + d)
CB	d	A6	RES	4, (IY + d)
CB	d	AE	RES	5, (IY + d)
CB	d	B6	RES	6, (IY + d)
CB	d	BE	RES	7, (IY + d)
CB	d	C6	SET	0, (IY + d)
CB	d	CE	SET	1, (IY + d)
CB	d	D6	SET	2, (IY + d)
CB	d	DE	SET	3, (IY + d)
CB	d	E6	SET	4, (IY + d)
CB	d	EE	SET	5, (IY + d)
CB	d	F6	SET	6, (IY + d)
CB	d	FE	SET	7, (IY + d)
E1			POP	IY
E3			EX	(SP), IY
E5			PUSH	IY
E9			JP	(IY)
F9			LD	SP, IY

<著者紹介>

日高　徹

1949年　栃木県宇都宮市生まれ
早稲田大学商学部卒業後、商社やカー用品メーカーに十数年勤務。現在はフリーのゲームデザイナー。
市販された作品に『ホーンテッドケーブ』『マジックガーデン』『北斗の拳』『ガンダーラ』など。著書は『マシン語ゲームプログラミング』『PC-8801シリーズ　マシン語サウンド・プログラミング』（以上アスキー）『PC-8800シリーズ　はじめてのマシン語』『PC-9800シリーズ　はじめてのマシン語』『Z80　マシン語秘伝の書』（以上啓学出版）『PC-9801シリーズ　マシン語ゲームグラフィックス』（小学館）。
趣味はトレーニング。剣道三段。運転免許証を全種類持つ、隠れプロドライバーでもある。

POPCOM BOOKS
PC-8801シリーズ　マシン語ゲームグラフィックス

1990年　5月　10日　初版第1刷発行
1991年　7月　10日　初版第3刷発行

著　者　日高　徹
発行人　相賀徹夫
発行所　小　学　館
〒101-01 東京都千代田区一ツ橋2-3-1
☎業務（03）3230-5333
☎販売（03）3230-5748 振替　東京8-200
編　集　株式会社　新企画社
〒101　東京都千代田区神田神保町3-3-7
昭和第2ビル　☎（03）3263-6940
印　刷　大日本印刷株式会社

レイアウト　㈱テクノネット・メディア・コミュニケーションズ
尾島由利子　井上恵美子　陳　蓓蕾　黒川昭子　鈴木美津子

ISBN4－09－385081－X

popcom

5インチ
2Dディスク
在中

CONTENTS

1.大型グラフィック用ツール
PMAN88

2.キャクタ・パターン用ツール
PTER88

3.スキャナコントロール用ツール
SCAN88

GAME GRAPHICS!

小学館